Basler racer

USER'S MANUAL FOR CAMERA LINK CAMERAS

Document Number: AW001185
Version: 01   Language: 000 (English)
Release Date: 13 June 2012

Preliminary

The information in this document is preliminary and all content is subject to change.

Applies to prototype cameras only.

BASLER
the power of sight

For customers in the U.S.A.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

You are cautioned that any changes or modifications not expressly approved in this manual could void your authority to operate this equipment.

The shielded interface cable recommended in this manual must be used with this equipment in order to comply with the limits for a computing device pursuant to Subpart J of Part 15 of FCC Rules.

For customers in Canada

This apparatus complies with the Class A limits for radio noise emissions set out in Radio Interference Regulations.

Pour utilisateurs au Canada

Cet appareil est conforme aux normes Classe A pour bruits radioélectriques, spécifiées dans le Règlement sur le brouillage radioélectrique.

Life Support Applications

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Basler customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Basler for any damages resulting from such improper use or sale.

Warranty Note

Do not open the housing of the camera. The warranty becomes void if the housing is opened.

Contacting Basler Support Worldwide

Europe:

Basler AG
An der Strusbek 60 - 62
22926 Ahrensburg
Germany

Tel.: +49-4102-463-515
Fax.: +49-4102-463-599

bc.support.europe@baslerweb.com

Americas:

Basler, Inc.
855 Springdale Drive, Suite 203
Exton, PA 19341
U.S.A.

Tel.: +1-610-280-0171
Fax.: +1-610-280-7608

bc.support.usa@baslerweb.com

Asia:

Basler Asia Pte. Ltd.
8 Boon Lay Way
# 03 - 03 Tradehub 21
Singapore 609964

Tel.: +65-6425-0472
Fax.: +65-6425-0473

bc.support.asia@baslerweb.com

www.baslerweb.com

Table of Contents

1 Specifications, Requirements, and Precautions

This chapter lists the camera models covered by the manual. It provides the general specifications for those models and the basic requirements for using them.

This chapter also includes specific precautions that you should keep in mind when using the cameras. We strongly recommend that you read and follow the precautions.

1.1 Models

The current Basler racer Camera Link® camera models are in the top row of the specification tables on the next pages of this manual. The camera models are differentiated by their sensor size and their maximum line rate at full resolution.

Unless otherwise noted, the material in this manual applies to all of the camera models listed in the tables. Material that only applies to a particular camera model or to a subset of models will be so designated.

1.2 General Specifications

Specification	raL2048-80km	raL4096-80km
Sensor Size	2048 pixels	4096 pixels
Sensor Type	Awaiba DR-2k-7 Monochrome Linear CMOS	Awaiba DR-4k-7 Monochrome Linear CMOS
Pixel Size	7 µm x 7 µm	
Max Line Rate	80 kHz	
Min Line Rate	No minimum when an external line trigger signal is used 100 Hz when an external line trigger signal is not used	
Mono/Color	Mono	
Sensor Bit Depth	12 bit	
Data Output Type	Camera Link base, medium or full configuration	
Connectors	One 6-pin Hirose micro-minature receptacle Two Camera Link 26-pin SDR connectors (also known as Mini Camera Link connectors)	
Camera Link Clock Speed	32.5 MHz, 48 MHz, 65 MHz, 83.5 MHz (selectable)	
Camera Link Tap Geometry	1X2, 1X4, 1X8, 1X10	
Pixel Bit Depths	8 bit, 10 bit, 12 bit	
Synchronization	Via external trigger signal, via software trigger signal or free run	
Exposure Control	Programmable via the camera API	
Power Requirements	PoCL (Power over Camera Link compliant) or +12 VDC (± 10 %), < 1 % ripple, supplied via the camera’s 6-pin connector.	
Max Power Consumption (at 12 VDC)	< 4 W	< 4.5 W
Lens Adapter	Universal camera front, suitable for C-mount (available as accessory)	
Size (L x W x H)	33.77 mm x 56 mm x 62 mm (without lens adapter or connectors) 48.57 mm x 56 mm x 62 mm (with C-mount lens adapter and connectors)	
Weight	~ 200 g (typical) without lens adapter ~ 230 g (typical) with C-mount lens adapter	
Conformity	CE, FCC, KCC, UL (in preparation), GenICam, GigE Vision, IP30	

Table 1: General Specifications - 2k and 4k Mono Cameras

1.3 Spectral Response

The following graph shows the quantum efficiency curve for monochrome cameras.

ⓘ	The quantum efficiency curve excludes lens characteristics and light source characteristics.

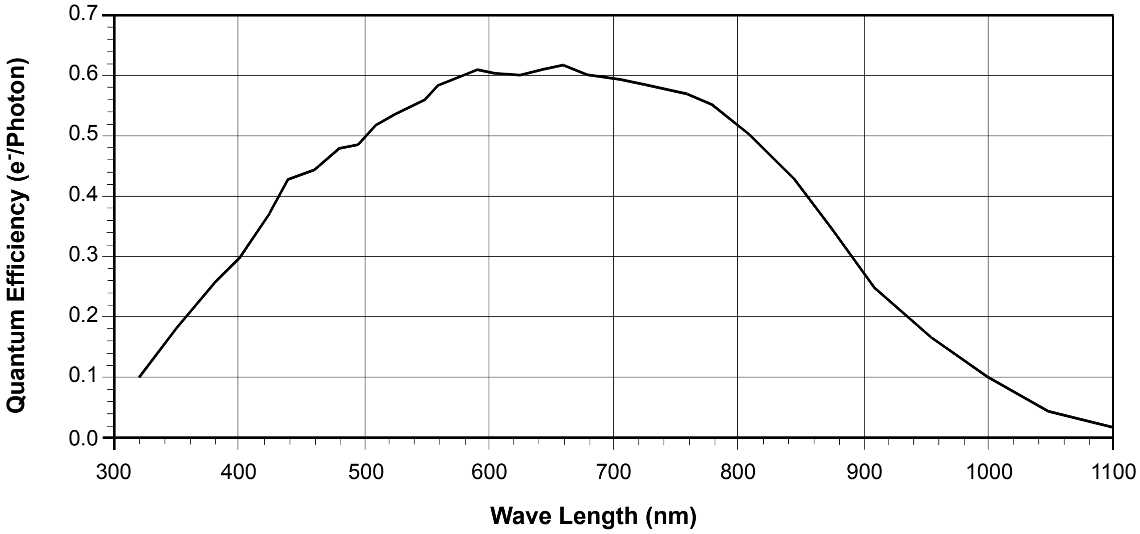


Fig. 1: Monochrome Camera Quantum Efficiency (From Sensor Data Sheet)

1.4 Mechanical Specifications

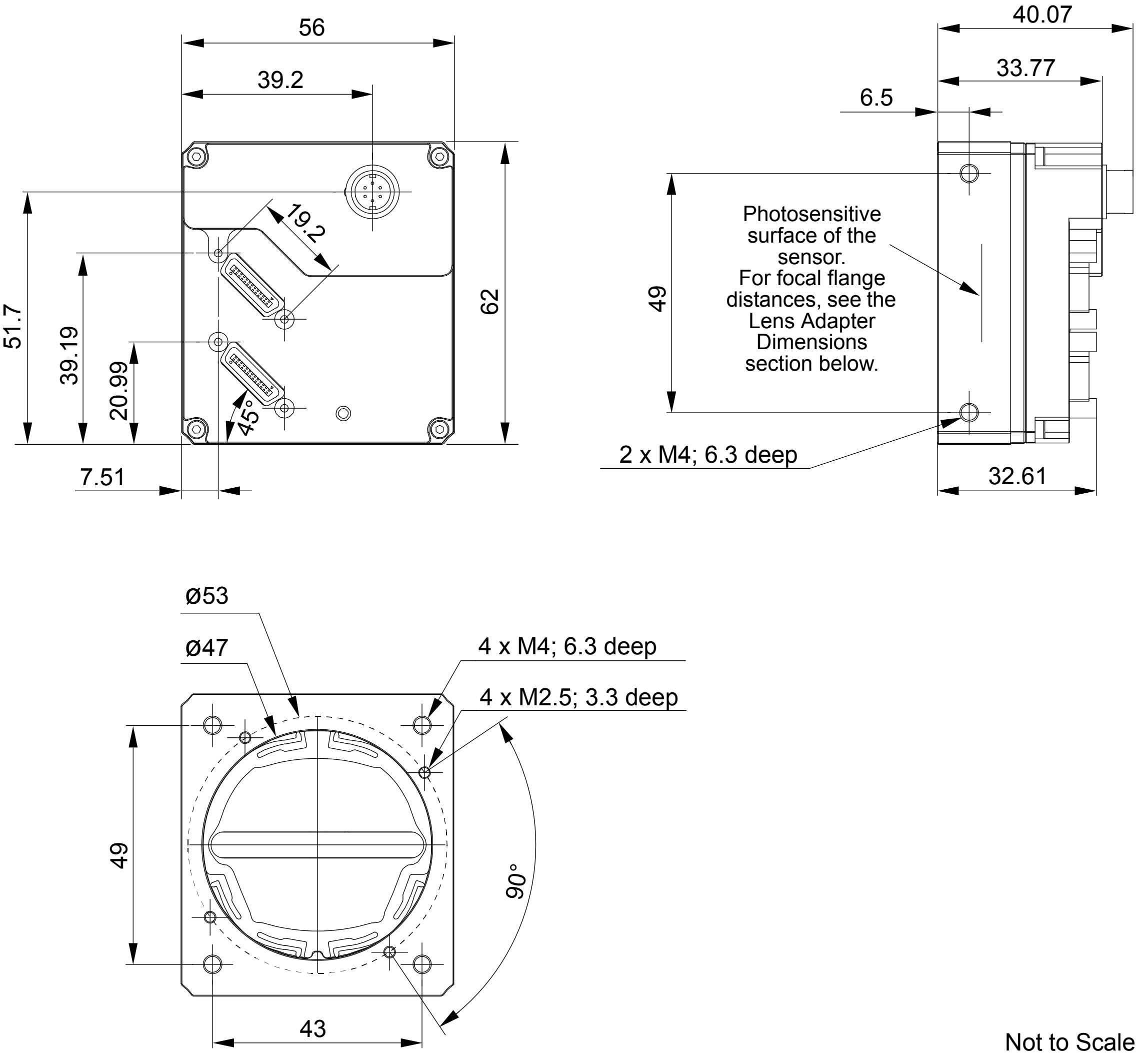
1.4.1 Camera Dimensions and Mounting Points

The cameras are manufactured with high precision. Planar, parallel, and angular sides guarantee precise mounting with high repeatability.

The camera housings conform to the IP30 protection class provided the camera front or the lens mount is covered by the protective plastic seal that is shipped with the camera.

The camera's dimensions in millimeters are as shown in the drawings below.

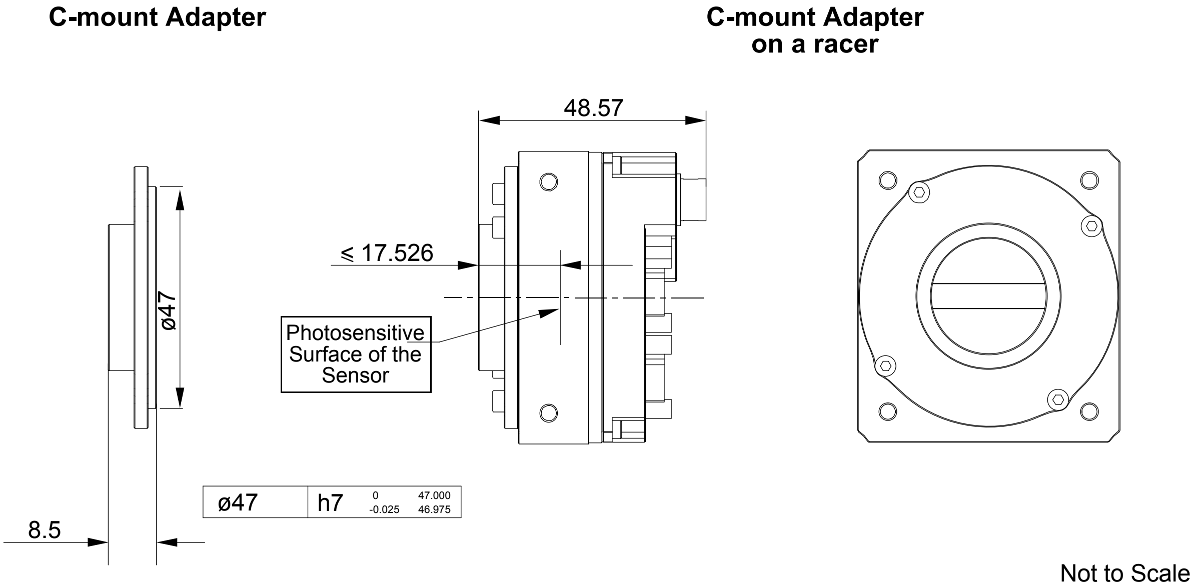
Camera housings are equipped with four mounting holes on the front and two mounting holes on each side as shown in the drawings.



Not to Scale

Fig. 2: Mechanical Dimensions (in mm)

1.4.2 Lens Adapter Dimensions



C-mount Adapter

C-mount Adapter on a racer

Not to Scale

Fig. 3: C-mount Adapter Dimensions

1.5 Software Licensing Information

The software in the camera includes the LWIP TCP/IP implementation. The copyright information for this implementation is as follows:

1.6 Avoiding EMI and ESD Problems

The cameras are frequently installed in industrial environments. These environments often include devices that generate electromagnetic interference (EMI) and they are prone to electrostatic discharge (ESD). Excessive EMI and ESD can cause problems with your camera such as false triggering or can cause the camera to suddenly stop capturing images. EMI and ESD can also have a negative impact on the quality of the image data transmitted by the camera.

To avoid problems with EMI and ESD, you should follow these general guidelines:

- Always use high quality shielded cables. The use of high quality cables is one of the best defenses against EMI and ESD.
- Try to use camera cables that are as short as possible and try to run the camera cables and power cables parallel to each other. Avoid coiling camera cables. If the cables are too long, use a meandering path rather then coiling the cables.
- Avoid placing camera cables parallel to wires carrying high-current, switching voltages such as wires supplying stepper motors or electrical devices that employ switching technology. **Placing camera cables near to these types of devices may cause problems with the camera.**
- Attempt to connect all grounds to a single point, e.g., use a single power outlet for the entire system and connect all grounds to the single outlet. This will help to avoid large ground loops. (Large ground loops can be a primary cause of EMI problems.)
- Use a line filter on the main power supply.
- Install the camera and camera cables as far as possible from devices generating sparks. If necessary, use additional shielding.
- Decrease the risk of electrostatic discharge by taking the following measures:
  - Use conductive materials at the point of installation (e.g., floor, workplace).
  - Use suitable clothing (cotton) and shoes.
  - Control the humidity in your environment. Low humidity can cause ESD problems.

ⓘ	The Basler application note called *Avoiding EMI and ESD in Basler Camera Installations* provides much more detail about avoiding EMI and ESD. This application note can be obtained from the camera’s Downloads section of our website: www.baslerweb.com

1.7 Environmental Requirements

1.7.1 Temperature and Humidity

Housing temperature during operation:	0 °C ... +50 °C (+32 °F ... +122 °F)
Humidity during operation:	20 % ... 80 %, relative, non-condensing
Storage temperature:	-20 °C ... +80 °C (-4 °F ... +176 °F)
Storage humidity:	20 % ... 80 %, relative, non-condensing

1.7.2 Heat Dissipation

You must provide sufficient heat dissipation to maintain the temperature of the camera housing at 50 °C or less. Since each installation is unique, Basler does not supply a strictly required technique for proper heat dissipation. Instead, we provide the following general guidelines:

- In all cases, you should monitor the temperature of the camera housing and make sure that the temperature does not exceed 50 °C. Keep in mind that the camera will gradually become warmer during the first hour of operation. After one hour, the housing temperature will have stabilized and will no longer increase.
- If your camera is mounted on a substantial metal component in your system, this may provide sufficient heat dissipation.
- The use of a fan to provide air flow over the camera is an extremely efficient method of heat dissipation. The use of a fan provides the best heat dissipation.

1.8 Precautions

NOTICE

Avoid dust on the sensor.

The camera is shipped with a protective plastic seal on the camera front or lens mount. To avoid collecting dust on the camera's sensor, make sure that you always put the protective seal in place when there is no lens mounted on the camera.

Also, make sure to always point the camera downward when there is no protective seal or lens on the camera front or lens mount.

NOTICE

Applying incorrect power can severely damage the camera.

1. If you supply camera power as Power over Camera Link (PoCL):
   - You must supply camera power in accordance with the Power over Camera Link (PoCL) specifications in the Camera Link standard.
   - You must use a PoCL compliant frame grabber.
   - You must use Camera Link cables that are specifically designed for PoCL as specified in the Camera Link standard.
2. If you supply camera power via the 6-pin receptacle:
   - You must supply camera power with the correct voltage: The camera‘s required nominal operating voltage is +12 VDC (±10 %), < 1% ripple, effective on the camera‘s connector.
   - You must supply camera power with the correct polarity.
   - You must avoid a voltage drop: If you supply camera power via a long cable a voltage drop can occur. We recommend that you provide +12 VDC separately through the wires connecting to pins 1 and 2 of the receptacle. We also recommend that you provide ground separately to the wires connecting to pins 5 and 6.

NOTICE

Making or breaking Camera Link connections incorrectly can severely damage the camera.

1. If you supply power to the camera via the Camera Link connection (PoCL), be sure that the power to the camera and to the frame grabber in your PC is switched off before you connect or disconnect the Camera Link cables.
2. If you supply power to the camera via the 6-pin connector, switch off the power to the connector before you connect or disconnect the Camera Link cables.
3. If you can’t switch off the power, be sure that:
   a. The plug for the 6-pin connector is the last plug that you plug into the camera when making connections.
   b. The plug for the 6-pin connector is the first plug that you unplug from the camera when breaking connections.

NOTICE

Incorrect plugs can damage the camera’s connectors.

The plug on the cable that you attach to the camera’s 6-pin connector must have 6 female pins.

NOTICE

Inappropriate code may cause unexpected camera behavior.

1. The code snippets provided in this manual are included as sample code only. Inappropriate code may cause your camera to function differently than expected and may compromise your application.
2. To ensure that the snippets will work properly in your application, you must adjust them to meet your specific needs and must test them thoroughly prior to use.
3. The code snippets in this manual are written in C++. Other programming languages can also be used to write code for use with Basler pylon. When writing code, you should use a programming language that is both compatible with pylon and appropriate for your application. For more information about the programming languages that can be used with Basler pylon, see the documentation included with the pylon package.

Warranty Precautions

To ensure that your warranty remains in force:

Do not remove the camera’s serial number label

If the label is removed and the serial number can’t be read from the camera’s registers, the warranty is void.

Do not open the camera housing

Do not open the housing. Touching internal components may damage them.

Keep foreign matter outside of the camera

Be careful not to allow liquid, flammable, or metallic material inside of the camera housing. If operated with any foreign matter inside, the camera may fail or cause a fire.

Avoid Electromagnetic fields

Do not operate the camera in the vicinity of strong electromagnetic fields. Avoid electrostatic charging.

Transport Properly

Transport the camera in its original packaging only. Do not discard the packaging.

Clean Properly

Avoid cleaning the surface of the camera’s sensor if possible. If you must clean it, use a soft, lint free cloth dampened with a small quantity of high quality window cleaner. Because electrostatic discharge can damage the sensor, you must use a cloth that will not generate static during cleaning (cotton is a good choice).

To clean the surface of the camera housing, use a soft, dry cloth. To remove severe stains, use a soft cloth dampened with a small quantity of neutral detergent, then wipe dry.

Do not use solvents or thinners to clean the housing; they can damage the surface finish.

Read the manual

Read the manual carefully before using the camera!

2 Software and Hardware Installation

The information you will need to do a quick, simple installation of the camera is included in the *Installation and Setup Guide for racer Camera Link Cameras* (AW001186xx000).

You can download the Installation and Setup Guide from the camera’s Downloads section of the Basler website: www.baslerweb.com

3 Tools for Changing Camera Parameters

This chapter provides an overview of the options available for changing the camera's parameters:

- The options available with the Basler pylon Driver Package let you change parameters and control the camera by using a stand-alone GUI (known as the pylon Viewer) or by accessing the camera from within your software application using the driver API.
- You can also control the camera and change parameters via direct access to the camera's register structure.

3.1 The pylon Driver Package

The Basler pylon Driver Package is designed to operate all Basler cameras that have an IEEE 1394a interface, an IEEE 1394b interface, or a GigE interface. The driver package also includes the pylon Camera Link Driver for use with newer Camera Link cameras.

The pylon drivers offer reliable, real-time image data transport into the memory of your PC at a very low CPU load.

For information about the features included in the pylon driver package and for information about installing pylon software, see the Installation and Setup Guide for Cameras Used with Basler's pylon API (AW000611xx000). You can download the document from the Basler website: www.baslerweb.com

For information about installing the pylon Camera Link Driver for use with racer Camera Link cameras, see the racer Installation and Setup Guide for Camera Link Cameras (AW001186xx000). You can download the guide from the camera's Downloads section of the website: www.baslerweb.com

The pylon package also includes several tools, in particular the pylon Viewer and the pylon API, that you can use to change the parameters on your camera. The remaining sections in this chapter provide an introduction to these tools.

3.1.1 The pylon Viewer

The pylon Viewer is included in Basler’s pylon Driver Package. The Viewer is a standalone application that lets you view and change most of the camera’s parameter settings via a GUI based interface. Using the pylon Viewer software is a very convenient way to get your camera up and running quickly when you are doing your initial camera evaluation or doing a camera design-in for a new project.

Note that with Camera Link cameras such as the racer, the viewer can not display the images captured by the camera. Most frame grabbers include software that will display the captured images, so you can use the pylon viewer to view or change the camera’s parameters and the frame grabber software to view the captured images.

For more information about using the viewer, see the racer Installation and Setup Guide for Camera Link Cameras (AW001186xx000). You can download the guide from the Basler website: www.baslerweb.com

3.1.2 The pylon API

After the pylon Driver Package has been installed on your PC, you can access all of the camera’s parameters and can control the camera’s full functionality from within your application software by using the pylon API. The pylon Programmer’s Guide and the pylon API Reference contain an introduction to the API and include information about all of the methods and objects included in the API. The programmer’s guide and API reference are included in the pylon SDK.

The Basler pylon Software Development Kit (SDK) includes a set of sample programs that illustrate how to use the pylon API to parameterize and operate the camera. These samples include Microsoft® Visual Studio® solution and project files demonstrating how to set up the build environment to build applications based on the API.

3.2 The Basler Binary Protocol Library

Basler racer Camera Link cameras have blocks of mapped memory space known as registers. By reading values from the registers, you can determine basic information about the camera and information about the camera’s current settings. By writing values to the registers, you can control how the camera’s features will operate.

If you use the Basler pylon software described in the previous section, the camera’s register structure is hidden. With pylon, a series of function calls allows you to change camera parameter settings without the need to know anything about the register that underlies each parameter.

If you desire, you can also change the camera parameter settings and control the camera by directly accessing the camera’s register structure. The Basler Binary Protocol Library (BBPL) provides functions that allow you to read data from or write data to the camera’s registers. The BBPL is an extension of the clALLSerial/clSerial API defined in Appendix B of the Camera Link Standard version 1.1 or higher. The BBPL adds convenience functions to this API that allow you to read from and write to the registers in Basler Camera Link cameras. The read and write requests are transmitted to the camera via a serial link between the camera and the frame grabber; the serial link is part of the standard Camera Link interface.

Sample code showing how to use the BBPL along with supporting documentation can be downloaded from the Basler website: www.baslerweb.com

When using the BBPL to change parameter values, you will need to know the details of the camera's register structure. For details of the register structure, refer to the document called Register Structure and Access Methods for Camera Link Cameras Used with Basler's pylon API (AW000895xx000). The document can be downloaded from the Basler website.

Note that if you are using an earlier Baser Camera Link camera that was originally designed to work with the Basler Binary Protocol II (e.g., the A400k, L400k, L800k, and the sprint), you can now use either the BBPL or the Binary Protocol II to access the camera's registers.

4 Camera Functional Description

This chapter provides an overview of the camera’s functionality from a system perspective. The overview will aid your understanding when you read the more detailed information included in the later chapters of the user’s manual.

Each camera employs a single line CMOS sensor chip designed for monochrome imaging. For 2k cameras, the sensor includes 2048 pixels. For 4k cameras, the sensor consists of two 2k sensor segments, resulting in a total of 4096 pixels.

Line start can be controlled by trigger signals transmitted to the camera via the Camera Link interface. Modes are available that allow the length of exposure time to be directly controlled by the line start signal or to be set for a pre-programmed period of time.

Accumulated charges are read out of the sensor when exposure ends. At readout, accumulated charges are moved from the sensor’s light-sensitive elements (pixels) into the analog processing section of the sensor (Fig. 4 on page 20). As the charges move from the pixels to the analog processing section, they are converted to voltages proportional to the size of each charge. The voltages from the analog processing section are next passed to a bank of 12 Bit Analog-to-Digital converters (ADCs).

Finally, the gray values pass through a section of the sensor where they receive additional digital processing and then they are moved out of the sensor. As each gray value leaves the sensor, it passes into an FPGA (Fig. 5 on page 20). Up to this point, all shifting is clocked according to the camera’s internal clock speed.

The gray values are then transmitted via the Camera Link interface to the frame grabber in your PC. Several different "Camera Link tap geometries" are available to determine how the pixel data will be transmitted over the Camera Link interface. The user can select the desired Camera Link tap geometry. See Section 7.2 on page 73 for more detailed information about Camera Link tap geometries.

The user can also select from several available pixel clock speeds for the Camera Link interface. See Section 8.1 on page 75 for more information about selecting the pixel clock speed.

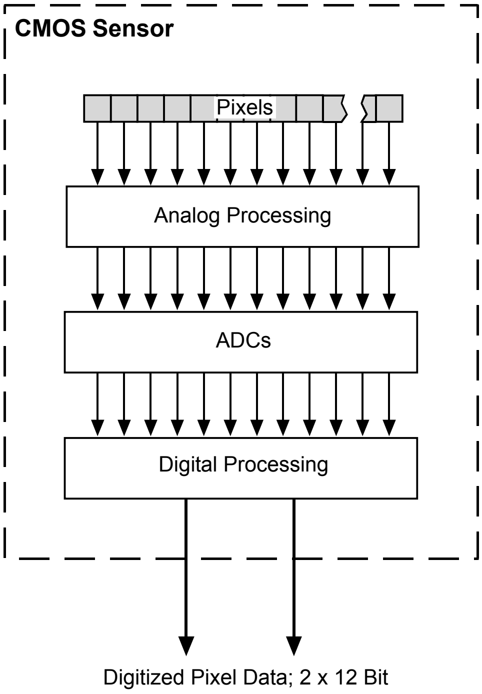


Fig. 4: CMOS Sensor Architecture for a 2k Sensor or 2k Sensor Segment with 7 µm x 7 µm Pixel Size

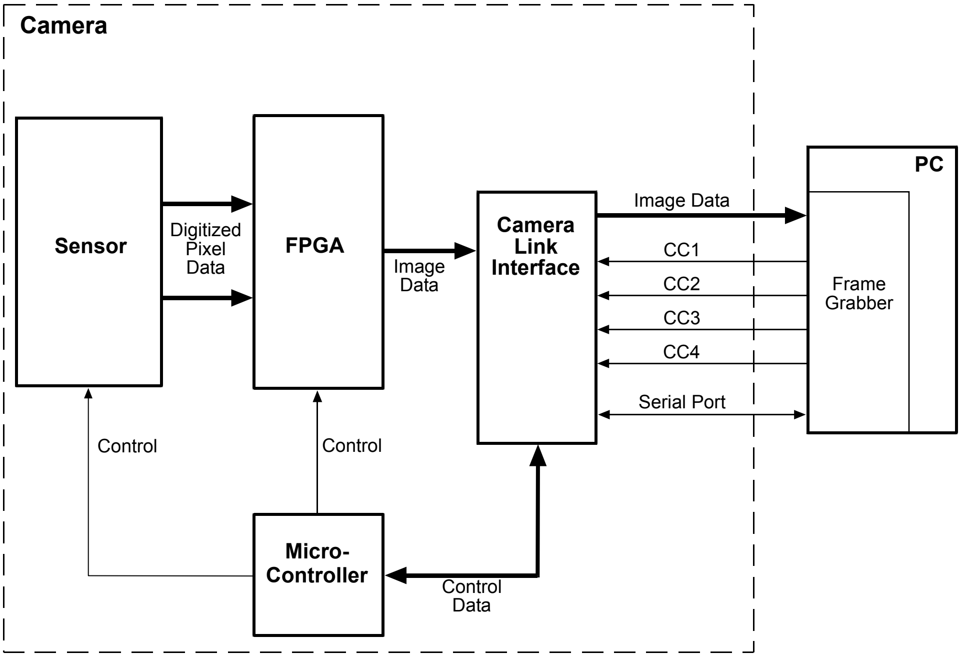


Fig. 5: Camera Block Diagram

5 Physical Interface

This chapter provides detailed information, such as pinouts and voltage requirements, for the physical interface on the camera. This information will be especially useful during your initial design-in process.

5.1 General Description of the Connections

The camera is interfaced to external circuity via connectors located on the back of the housing:

- Two 26-pin, 0.03” pin spacing, Shrunk Delta Ribbon (SDR) female connectors used to transmit video data, control signals, and configuration commands. (This type of connector is also known as a Mini Camera Link connector).

  The **base** Camera Link connector can also be used to supply power to the camera in accordance with the Power over Camera Link (PoCL) specifications in the Camera Link standard.
- One 6-pin receptacle used to provide power to the camera if PoCL is not used.

There is also a LED indicator located on the back of the camera as shown in the drawing.

The drawing below shows the location of the three connectors and the LED indicator.

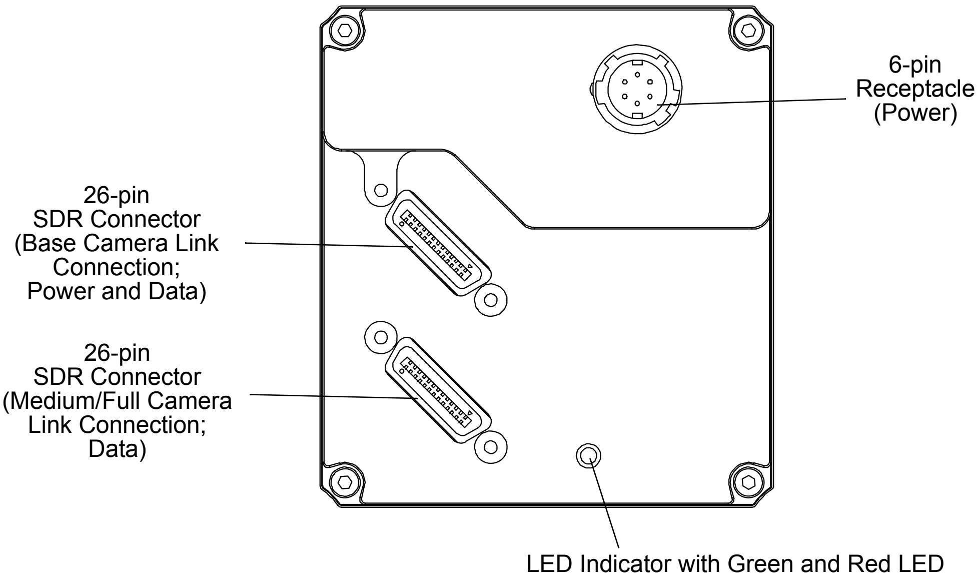


Fig. 6: Camera Connectors and LED Indicator

5.2 Camera Connector Pin Assignments and Numbering

5.2.1 6-pin Receptacle

The 6-pin receptacle is used to supply power to the camera. The pin assignments and pin numbering for the receptacle are as shown in Table 2.

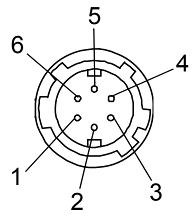
	Pin	Designation
	1	+12 VDC (± 10%), < 1 % ripple, Camera Power *
	2	+12 VDC (± 10%), < 1 % ripple, Camera Power *
	3	Not Connected
	4	Not Connected
	5	DC Ground **
	6	DC Ground **

Table 2: Pin Assignments and Numbering for the 6-pin Receptacle

* Pins 1 and 2 are tied together inside of the camera.
** Pins 5 and 6 are tied together inside of the camera.

5.2.2 26-pin SDR Connectors

Two 26-pin, 0.03” pin spacing, Shrunk Delta Ribbon (SDR) female connectors are used to transmit video data, control signals, and configuration commands.

The **base** Camera Link SDR connector can also be used to supply power to the camera in accordance with the Power over Camera Link (PoCL) specifications in the Camera Link standard.

The pin assignments and pin numbering are shown in Table 3 on page 23 for the base Camera Link SDR connector and in Table 4 on page 24 for the medium/full Camera Link SDR connector.

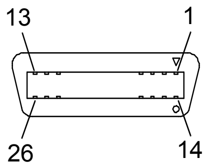
Pin Number	Signal Name	Direction	Level	Function
1, 26 *	Cam Pow.	In	+12 VDC	Camera power, +12 VDC (±10%), < 1% ripple. For more details about PoCL power requirements see the Camera Link specifications v. 2.0 and above.
13, 14 **	Power Ret.		Return	Camera power return (Gnd)
2	X0-	Output	Camera Link LVDS	Data from transmitter circuit X
15	X0+			
3	X1-	Output	Camera Link LVDS	Data from transmitter circuit X
16	X1+			
4	X2-	Output	Camera Link LVDS	Data from transmitter circuit X
17	X2+			
6	X3-	Output	Camera Link LVDS	Data from transmitter circuit X
19	X3+			
5	XClk-	Output	Camera Link LVDS	Pixel clock from transmitter circuit X
18	XClk+			
7	SerTC+	Input	RS-644 LVDS	Serial communication data receive (SerTC = "serial to camera")
20	SerTC-			
8	SerTFG-	Output	RS-644 LVDS	Serial communication data transmit (SerTFG = "serial to frame grabber")
21	SerTFG+			
9	CC1-	Input	RS-644 LVDS	Configurable
22	CC1+			
10	CC2+	Input	RS-644 LVDS	Configurable
23	CC2-			
11	CC3-	Input	RS-644 LVDS	Configurable
24	CC3+			
12	CC4+	Input	RS-644 LVDS	Configurable
25	CC4-			
		* Pins 1 and 26 are tied together in the camera. ** Pins 13 and 14 are tied together in the camera.		

Table 3: Pin Assignments and Numbering for the Base Configuration 26-pin SDR Connector

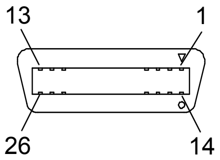
Pin Number	Signal Name	Direction	Level	Function
1, 26 *	Cam Pow.	In	+12 VDC	Not used
13, 14 **	Power Ret.		Return	Gnd
2	Y0-	Output	Camera Link LVDS	Data from transmitter circuit Y
15	Y0+			
3	Y1-	Output	Camera Link LVDS	Data from transmitter circuit Y
16	Y1+			
4	Y2-	Output	Camera Link LVDS	Data from transmitter circuit Y
17	Y2+			
6	Y3-	Output	Camera Link LVDS	Data from transmitter circuit Y
19	Y3+			
5	YClk-	Output	Camera Link LVDS	Pixel clock from transmitter circuit Y
18	YClk+			
7	T+			Connected to T- with 100R; not used
20	T-			Connected to T+ with 100R; not used
8	Z0-	Output	Camera Link LVDS	Data from transmitter circuit Z
21	Z0+			
9	Z1-	Output	Camera Link LVDS	Data from transmitter circuit Z
22	Z1+			
10	Z2-	Output	Camera Link LVDS	Data from transmitter circuit Z
23	Z2+			
12	Z3-	Output	Camera Link LVDS	Data from transmitter circuit Z
25	Z3+			
11	ZClk-	Output	Camera Link LVDS	Pixel clock from transmitter circuit Z
24	ZClk+			
	* Pins 1 and 26 are tied together in the camera. ** Pins 13 and 14 are tied together in the camera.			

Table 4: Pin Assignments and Numbering for the Medium/Full Configuration 26-pin SDR Connector

5.3 Connector Types

5.3.1 6-pin Connector

The 6-pin connector on the camera is a Hirose micro receptacle (part number HR10A-7R-6PB) or the equivalent.

The recommended mating connector is the Hirose micro plug (part number HR10A-7P-6S) or the equivalent.

5.3.2 26-pin SDR Connectors

The 26-pin connectors on the camera are female, 0.03 inch pin spacing, SDR connector as called for in the Camera Link specification.

The recommended mating connector is also defined in the Camera Link Specification.

ⓘ	The "SDR" (Shrunk Delta Ribbon) designation is the naming used by the 3M company. Other companies use different names for this type of connector. For example, Honda uses "HDR" as the connector name.

5.4 Cabling Requirements

5.4.1 Power Cable

A single power cable is used to supply auxiliary power to the camera.

The end of the power cable that connects to the camera's 6-pin connector must be terminated with a Hirose micro plug (part number HR10A-7P-6S) or the equivalent. The cable must be wired as shown in Fig. 7.

For proper EMI protection, the power cable terminated with the Hirose connector and attached to the camera must be a twin-cored, shielded cable. Also, the Hirose plug must be connected to the cable shield and the shield must be connected to earth ground at the power supply.

Close proximity to strong electromagnetic fields should be avoided.

NOTICE

An incorrect plug can damage the 6-pin connector.

The plug on the cable that you attach to the camera's 6-pin connector must have 6 female pins.

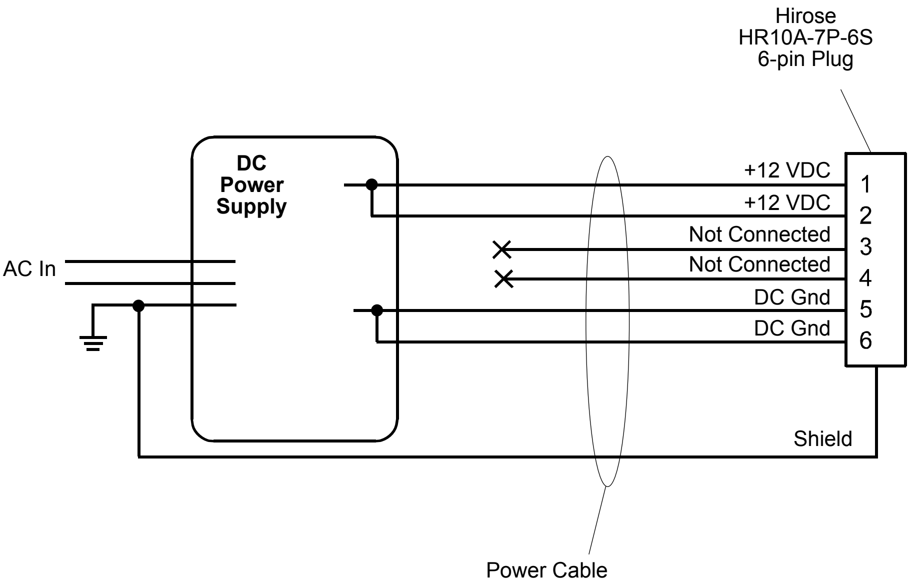


Fig. 7: Power Cable

5.4.2 Camera Link Cables

The Camera Link cables must meet the Mini Camera Link cable specifications specified in Appendix D of the Camera Link Standard.

Close proximity to strong electromagnetic fields should be avoided.

ⓘ	Note: The maximum recommended Camera Link cable length depends on Camera Link clock speed and Camera Link tap geometry. For optimum performance of the Camera Link interface, make sure not to exceed the applicable maximum recommended Camera Link cable length.

5.5 Camera Power

Power can be supplied to the camera in either one of two ways:

- Via the base Camera Link SDR connector (Power over Camera Link).
- Via the 6-pin connector (auxiliary power).

5.5.1 Supplying Power Over Camera Link

Power can be supplied to the camera via a Camera Link cable as specified in the Camera Link standard v. 2.0 and above. This method of supplying power to the camera is known as Power over Camera Link or PoCL. PoCL camera power can **only** be supplied via the base Camera Link SDR connector.

The required nominal operating voltage for the camera is +12 VDC (± 10%) with less than one percent ripple. For more details about PoCL power requirements, see the Camera Link specifications v. 2.0 and above. Power consumption is as shown in the specification tables in Section 1 of this manual.

The power for the camera is supplied by the connection to your frame grabber. Therefore, if you are planning to use PoCL to power the camera, your frame grabber must be PoCL compliant.

NOTICE

If you are supplying power to the camera using PoCL:

1. The camera’s required nominal operating voltage is +12 VDC (± 10%), effective on the camera’s connector.
2. Failure to use a Power over Camera Link compliant frame grabber can result in severe damage to the camera.
3. Failure to use a Camera Link cable that is specifically designed for PoCL (as specified in the Camera Link standard) can result in severe damage to the camera.

NOTICE

Making or breaking Camera Link connections incorrectly can cause severe damage to the camera.

If you are supplying power to the camera via the Camera Link connection (PoCL), be sure that the power to the camera and to the frame grabber in your PC is switched off before you connect or disconnect the Camera Link cables.

5.5.2 Supplying Auxiliary Power Via the 6-pin Connector

Auxiliary power can be supplied to the camera via the 6-pin connector on the back of the camera. Nominal operating voltage is +12 VDC (± 10%) with less than one percent ripple. Power consumption is as shown in the specification tables in Section 1 of this manual.

Close proximity to strong magnetic fields should be avoided.

See Section 5.2.1 on page 22 for a description of the connector pinouts.

See Section 5.4.1 on page 26 for a description of the power cable to be used with the 6-pin connector.

NOTICE

- Applying incorrect power can damage the camera.

  The camera‘s required nominal operating voltage is +12 VDC (±10 %), effective on the camera‘s connector.
- Applying power with the wrong polarity can severely damage the camera.

  Make sure that the polarity of the power applied to the camera is correct. Applying power with the wrong polarity can severely damage the camera.

NOTICE

An incorrect plug can damage the 6-pin connector.

The plug on the cable that you attach to the camera’s 6-pin connector must have 6 female pins.

NOTICE

Making or breaking Camera Link connections incorrectly can severely damage the camera.

1. If you supply power to the camera via the 6-pin connector, switch off the power to the connector before you connect or disconnect the Camera Link cables.
2. If you can’t switch off the power, be sure that:
   a. The plug for the 6-pin connector is the last plug that you plug into the camera when making connections.
   b. The plug for the 6-pin connector is the first plug that you unplug from the camera when breaking connections.

5.5.3 LED Indicator

The LED indicator on the back of the camera signals whether power is present and also provides some basic error indications for the camera.

For more information, see Section 8.9 on page 98.

5.6 I/O in the Camera Link Interface

5.6.1 Inputs

The camera is equipped with four input lines built into the Camera Link interface. These lines are designated as CC1, CC2, CC3, and CC4 as specified in the Camera Link standard. Typically, input signals are applied to these lines by the frame grabber board attached to the camera. The frame grabber board can typically be configured to supply different types of signals to these inputs as required by the camera user.

On the camera side, you can select these inputs to act as the source signal for the line start trigger signals.

For more information about using CC1, CC2, CC3, and CC4 on the specific type of frame grabber installed in your system, refer to the documentation for you frame grabber board.

You can also obtain some general information about how these lines are implemented in the Camera Link interface from the Basler document named racer Camera Link Information for Frame Grabber Designers (AW001187xx000). You can obtain the document from the Basler website:

www.baslerweb.com

For more information about using CC1, CC2, CC3, and CC4 as the source signal for a camera function, see Section 6.1 on page 43.

5.6.1.1 Input Line Debouncers

Each individual input line is equipped with a debouncer. The debouncer aids in discriminating between valid and invalid input signals and only lets valid signals pass to the camera. The debouncer value specifies the minimum time that an input signal must remain high or remain low in order to be considered a valid input signal.

ⓘ	We recommend setting the debouncer value so that it is slightly greater than the longest expected duration of an invalid signal. Setting the debouncer to a value that is too short will result in accepting invalid signals. Setting the debouncer to a value that is too long will result in rejecting valid signals.

Note that the debouncer creates a delay between the arrival of a valid signal at the camera and its transfer to the camera’s internal circuitry. The duration of the delay will be determined by the debouncer value.

Fig. 8 illustrates how the debouncer filters out invalid input signals, i.e. signals that are shorter than the debouncer value. The diagram also illustrates how the debouncer delays a valid signal.

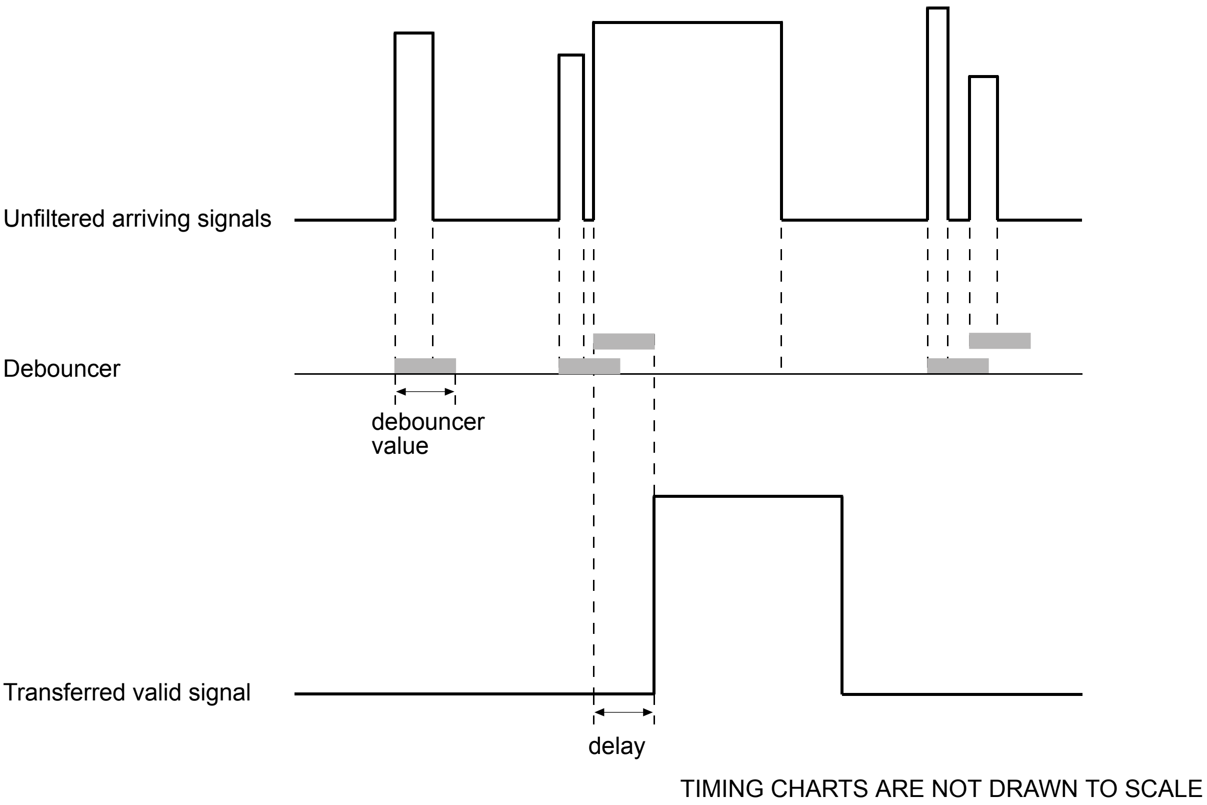

Unfiltered arriving signals
Debouncer
debouncer value
Transferred valid signal
delay
TIMING CHARTS ARE NOT DRAWN TO SCALE

Fig. 8: Filtering of Input Signals by the Debouncer

Setting the Debouncer Using Basler pylon

The debouncer value is determined by the value of the Line Debouncer Time Abs parameter. The parameter is set in microseconds and can be set in a range from 0 to approximately 1 s.

To set the debouncer:

- Use the Line Selector to select CC1, CC2, CC3, or CC4.
- Set the value of the Line Debouncer Time Abs parameter.

You can set the Line Selector and the value of the Line Debouncer Abs parameter from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Select the CC1 line
Camera.LineSelector.SetValue( LineSelector_CC1 );

// Set the parameter value to 150 microseconds
Camera.LineDebouncerTimeAbs.SetValue( 150 );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting the Debouncer Using Direct Register Access

To set the value of the input line debouncers via direct register access:

- For the CC1 line, set the value of the Input Debouncer Time CC1 register as desired (the value represents milliseconds).
- For the CC2 line, set the value of the Input Debouncer Time CC2 register.
- For the CC3 line, set the value of the Input Debouncer Time CC3 register.
- For the CC4 line, set the value of the Input Debouncer Time CC4 register.

For more information about direct register access, see Section 3.2 on page 16.

5.6.1.2 Input Line Inverters

You can set CC1, CC2, CC3, and the CC4 line to invert or not to invert the input signal.

Setting an Input Line for Invert Using Basler pylon

To set the invert function on an input line:

- Use the Line Selector to select CC1, CC2, CC3, or CC4.
- Set the value of the Line Inverter parameter to true to enable inversion on the selected line or to false to disable inversion.

You can set the Line Selector and the Line Inverter parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Select the CC1 line
Camera.LineSelector.SetValue( LineSelector_CC1 );
// Disable the inverter on the selected line
Camera.LineInverter.SetValue( false );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting an Input Line for Invert Using Direct Register Access

To set the invert function on an input line via direct register access:

- For the CC1 line, set the value of the Line Inverter CC1 register to 0 (false) or 1 (true) as desired.
- For the CC2 line, set the value of the Line Inverter CC2 register.
- For the CC3 line, set the value of the Line Inverter CC3 register.
- For the CC4 line, set the value of the Line Inverter CC4 register.

For more information about direct register access, see Section 3.2 on page 16.

5.6.1.3 Selecting an Input Line as a Source Signal for a Camera Function

You can select an input line as the source signal for the following camera function:

- the Line Start Trigger

Note that to use an input line as the source signal for a camera function, you must apply a signal to the input line that is appropriately timed for the function.

For detailed information about selecting an input line as the source signal for the camera's Line Start Trigger function, see Section 6.1 on page 43.

Default Input Line Selection

Line CC1 is selected as the source signal for the camera's Line Start Trigger function.

5.6.2 Outputs

As specified in the Camera Link standard, a "CL Spare" data bit is included in the Camera Link interface. On racer Camera Link cameras, the CL Spare data bit can be used as a camera output line.

You can select any one of the camera’s standard output signals to act as the source signal for the CL Spare output line. The camera has these standard output signals available:

- Line Trigger Wait
- Exposure Active
- User Output
- Frequency Converter

You can also designate the line as "user settable". If the output line is designated as user settable, you can use the camera’s API to set the state of the line as desired.

> The CL Spare data bit is not available when the camera is used with either of these Camera Link tap geometries:
>
> - 1X8 Camera Link tap geometry used with the Mono 10 pixel format
> - 1X10 Camera Link tap geometry.
>
> The CL Spare data bit is not directly accessible by the camera user. The data bit must be accessed via the frame grabber attached to the camera. Not all frame grabbers provide users with direct access to this bit. Consult your frame grabber supplier for more information.

For more information about using the CL Spare bit on your specific type of frame grabber installed in your system, refer to the documentation for you frame grabber board.

You can also obtain some general information about how this bit is implemented in the Camera Link interface from the Basler document named racer Camera Link Information for Frame Grabber Designers (AW001187xx000). You can obtain the document from the Basler website: www.baslerweb.com

5.6.2.1 Minimum Output Pulse Width

You can use the minimum output pulse width feature to ensure that even very narrow camera output signals will reliably be detected by other devices. The feature allows you to set the output signal to a minimum width.

Setting the Minimum Output Pulse Width Using Basler pylon

The MinOutPulseWidthAbs parameter sets the CL Spare output signals to a minimum width. The parameter is set in microseconds and can be set in a range from 0 to 100 µs

You can set the Line Selector and the value of the MinOutPulseWidthAbs parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Select the output line
Camera.LineSelector.SetValue(LineSelector_CLSpare);

// Set the parameter value to 10.0 microseconds
Camera.MinOutPulseWidthAbs.SetValue(10.0);
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about selecting the source signal for an output line on the camera, see Section 5.6.2.3 on page 37.

Setting the Minimum Output Pulse Width Using Direct Register Access

To set the value for the minimum output pulse width via direct register access:

Set the value of the Min Out Pulse Width CL Spare register as desired (the value represents microseconds).

For more information about direct register access, see Section 3.2 on page 16.

5.6.2.2 Output Line Inverters

You can set the CL Spare output line to invert or not to invert the input signal.

Setting the CL Spare Output Line for Invert Using Basler pylon

To set the invert function on the CL Spare output line:

- Use the Line Selector to select CL Spare.
- Set the value of the Line Inverter parameter to true to enable inversion on the selected line or to false to disable inversion.

You can set the Line Selector and the Line Inverter parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Select the CL Spare output line
Camera.LineSelector.SetValue( LineSelector_CLSpare );
// Disable the inverter on the selected line
Camera.LineInverter.SetValue( false );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting the CL Spare Output Line for Invert Using Direct Register Access

To set the invert function on the CL Spare output line via direct register access:

- For the CL Spare output line, set the value of the Line Inverter CL Spare register to 0 (false) or 1 (true) as desired.

For more information about direct register access, see Section 3.2 on page 16.

5.6.2.3 Selecting the Source Signal for an Output Line

To make an output line useful, you must select a source signal for the output line.

You can select any one of the camera’s standard output signals to act as the source signal for the CL Spare output line. The camera has these standard output signals available:

- Line Trigger Wait
- Exposure Active

You can also select one of the following as the source signal for the output:

- the "User Output" signal (when you select "user output" as the source signal for the CL Spare output line, you can use the camera’s API to set the state of the line as you desire)
- Off (when "off" is selected as the source signal, the output is disabled.)

Selecting the Source Signal Using Basler pylon

To select one of the camera's standard output signals as the source signal for the CL Spare output line or to select user output or off:

- Use the Line Selector to select CL Spare as the output line.
- Set the value of the Line Source Parameter to Line Trigger Wait, Exposure Active, User Output, or Off. This will select the source signal for the line.

You can set the Line Selector and the Line Source parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Select the line trigger wait signal for the CL Spare output line
Camera.LineSelector.SetValue( LineSelector_CLSpare );
Camera.LineSource.SetValue( LineSource_LineTriggerWait );

// Select the exposure active signal for the CL Spare output line
Camera.LineSelector.SetValue( LineSelector_CLSpare );
Camera.LineSource.SetValue( LineSource_ExposureActive );

// Select the CL Spare output line as a user output
Camera.LineSelector.SetValue( LineSelector_CLSpare );
Camera.LineSource.SetValue( LineSource_UserOutput );

// Disable the CL Spare output line
Camera.LineSelector.SetValue( LineSelector_CLSpare );
Camera.LineSource.SetValue( LineSource_Off );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 3.1 on page 15.

For more information about the Line Trigger Wait signal, see Section 6.3.3 on page 64.

For more information about the Exposure Active signal, see Section 6.3.1 on page 62.

For more information about working with outputs that have "user settable" as the signal source, see Section 5.6.2.4 on page 39.

Selecting the Source Signal Using Direct Register Access

To select a camera output signal as the source signal for the CL Spare output line or to designate the line as user settable via direct register access:

- Set the value of the Line Source CL Spare register to Line Trigger Wait, Exposure Active, or User as desired.

For more information about direct register access, see Section 3.2 on page 16.

Default Output Line Source Signal Selections

By default, the camera’s Exposure Active signal is selected as the source signal for the CL Spare output line.

5.6.2.4 Setting the State of a User Settable Output Line

As mentioned in the previous section, you can select "user output" as the signal source for the CL Spare output line. For the output line that has "user output" as the signal source, you can use camera parameters to set the state of the line.

ⓘ If you have the invert function enabled on the output line that is designated as a user output, the user setting sets the state of the line before the inverter.

Setting the State of a User Output Line Using pylon

To set the state of a single user output line:

- Use the User Output Selector to select CL Spare as the output line you want to set.
- Set the value of the User Output Value parameter to true (high) or false (low). This will set the state of the selected line.

You can set the Output Selector and the User Output Value parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to select "user settable" as the source signal for output line 2 and how to set the state of the output line:

```
// Select "user output" as the signal source for the CL Spare output line
Camera.LineSelector.SetValue( LineSelector_CLSpare );
Camera.LineSource.SetValue( LineSource_UserOutput );

//Set the state of the CL Spare output line and then read the state
Camera.UserOutputSelector.SetValue( UserOutputSelector_CLSpare );
Camera.UserOutputValue.SetValue( true );
bool currentUserOutput2State = Camera.UserOutputValue.GetValue( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

Setting the State of a User Output Line Using Direct Register Access

To set the state of a user settable output line via direct register access:

- Set the value of the User Output CL Spare register to 1 (true) or 0 (false) as desired.

For more information about direct register access, see Section 3.2 on page 16.

5.7 Checking the State of the I/O Lines

5.7.1 Checking the State of a Single Line

Checking the State of an I/O Line Using Basler pylon

You can determine the current state of each I/O line. To check the state of a line:

- Use the Line Selector parameter to select a line.
- Read the value of the Line Status parameter to determine the current state of the line. A value of true means the line's state is currently high and a value of false means the line's state is currently low.

You can set the Line Selector and read the Line Status parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector and read the parameter value:

```
// Select the CC1 line and read the state
Camera.LineSelector.SetValue( LineSelector_CC1 );
bool CC1 = Camera.LineStatus.GetValue( );

// Select the CLSpare line and read the state
Camera.LineSelector.SetValue( LineSelector_ClSpare );
bool ClSpareState = Camera.LineStatus.GetValue( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Checking the State of an I/O Line Using Direct Register Access

To check the current state of an I/O line via direct register access:

- For the CC1 line, read the value of the Line Status CC1 register. The value will indicate 1 (true) or 0 (false).
- For the CC2 line, read the value of the Line Status CC2 register.
- For the CC3 line, read the value of the Line Status CC3 register.
- For the CC4 line, read the value of the Line Status CC4 register.
- For the CL Spare line, read the value of the Line Status CL Spare register.

For more information about direct register access, see Section 3.2 on page 16.

5.7.2 Checking the State of All Lines

Checking the State Using Basler pylon

You can determine the current state of all input and output lines by reading the value of the Line Status All parameter. You can read the Line Status All parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to read the parameter value:

```
int64_t lineState = Camera.LineStatusAll.GetValue( );
```

The Line Status All parameter is a 32 bit value. As shown in Fig. 9, certain bits in the value are associated with each I/O line, and each of these bits will indicate the state of the associated line. If a bit is 0, it indicates that the state of the associated line is currently low. If a bit is 1, it indicates that the state of the associated line is currently high.

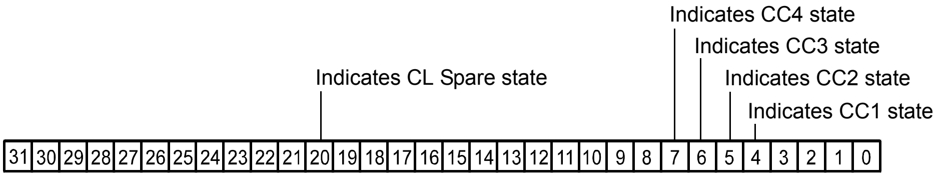


Fig. 9: Line Status All Parameter Bits

You can also use the Basler pylon Viewer application to easily read the parameter.

Checking the State Using Direct Register Access

To check the current state of all I/O lines via direct register access, read the value of the Line Status All register. The register holds a 32 bit value that indicates the state of each I/O line. The mapping of the bits in the value to I/O lines is similar to the mapping described above for access via Basler pylon.

For more information about direct register access, see Section 3.2 on page 16.

5.8 Checking the Line Logic of the I/O Lines

Checking the Line Logic Using Basler pylon

You can determine the type of line logic for each I/O line using Basler pylon:

- Use the Line Selector parameter to select a line.
- Read the value of the Line Logic parameter to determine the type of line logic used by the line. The parameter will indicate whether the logic is positive or negative.

You can set the Line Selector and read the Line Logic parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector and read the parameter value:

```
// Select the CC1 line and read the line logic type
Camera.LineSelector.SetValue( LineSelector_CC1 );
LineLogicEnums lineLogicCC1 = Camera.LineLogic.GetValue( );

// Select the CLSpare line and read line logic type
Camera.LineSelector.SetValue( LineSelector_ClSpare );
LineLogicEnums lineLogicClSpare = Camera.LineLogic.GetValue( );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Checking the Line Logic Using Direct Register Access

To check the Line Logic of an I/O line via direct register access:

- For the CC1 line, read the value of the Line Logic CC1 register. The value will indicate 1 (positive) or 0 (negative).
- For the CC2 line, read the value of the Line Logic CC2 register.
- For the CC3 line, read the value of the Line Logic CC3 register.
- For the CC4 line, read the value of the Line Logic CC4 register.
- For the CL Spare line, read the value of the Line Logic CL Spare register.

For more information about direct register access, see Section 3.2 on page 16.

6 Acquisition Control

This section provides detailed information about controlling the acquisition of image information. You will find details about triggering line acquisition, about setting the exposure time for acquired lines, about setting the camera's line acquisition rate, and about how the camera's maximum allowed line acquisition rate can vary depending on the current camera settings.

The following elements are involved in controlling the acquisition of lines:

- the line start trigger
- exposure time control

6.1 Line Start Triggering

The line start trigger is used to start a line acquisition. Keep in mind that the camera will only react to a line start trigger when the line start trigger is valid. If the line start trigger is invalid, line start triggers will be ignored.

The first parameter associated with the line start trigger is the Trigger Mode parameter. The Trigger Mode parameter has two available settings: off and on.

6.1.1 Line Start with Trigger Mode = Off

When the Trigger Mode parameter is set to off, selection of a source signal for the line start trigger is not required. With the trigger mode set to off, the camera operates the line start trigger automatically. The rate at which the line start triggers are generated will be determined by the camera's Acquisition Line Rate Abs parameter:

- If the parameter is set to a value less than the maximum allowed line acquisition rate, the camera will generate triggers at the rate specified by the parameter setting.
- If the parameter is set to a value greater than the maximum allowed line acquisition rate, the camera will generate line start triggers at the maximum allowed line rate.

For more information about the maximum allowed line rate, see Section 6.4 on page 66.

Exposure Time Control

When the trigger mode is set to off, the exposure time for each line acquisition is determined by the value of the camera’s Exposure Time parameter.

The minimum and the maximum allowed exposure time for each acquired line are as shown in Table 5.

	raL2048-80km	**raL4096-80km**
Min	2.0 µs	2.0 µs
Max	10000 µs	10000 µs

Table 5: Minimum and Maximum Allowed Exposure Times

If you are using a GenICam compliant tool such as the Basler pylon Viewer and you attempt to set the exposure time to exactly the minimum allowed or to exactly the maximum allowed, you will see unusual error codes. This is an artifact of a rounding error in the GenICam interface architecture. As a work around, you could set the exposure time slightly above the minimum or below the maximum. Values between the minimum and the maximum are not affected by the problem.

For more information about setting exposure time, see Section 6.1.3 on page 46.

6.1.2 Line Start with Trigger Mode = On

When Trigger Mode is set to on, you must select a source signal for the line start trigger. Trigger Source specifies the source signal. The available selections for Trigger Source are:

- Software - When the line start trigger source is set to software, the user triggers line start by issuing a TriggerSoftware command to the camera from the host PC. Each time a TriggerSoftware command is received by the camera, the line start trigger will become valid. It will become invalid during line acquisition and will become valid again when the next TriggerSoftware command is received and when the camera is ready again for a new line acquisition.
- CC1 - When the source signal is set to CC1, you apply a line start trigger signal to the camera by injecting an externally generated electrical signal (a "hardware trigger") into CC1 in the Camera Link interface.
- CC2 - When the source signal is set to CC2, you apply a line start trigger signal to the camera by injecting an externally generated electrical signal (a "hardware trigger") into CC2 in the Camera Link interface.
- CC3 - When the source signal is set to CC3, you apply a line start trigger signal to the camera by injecting an externally generated electrical signal (a "hardware trigger") into CC3 in the Camera Link interface.
- CC4 - When the source signal is set to CC4, you apply a line start trigger signal to the camera by injecting an externally generated electrical signal (a "hardware trigger") into CC4 in the Camera Link interface.

If Trigger Source is set to CC1, CC2, CC 3, or CC4, the user must also set the Trigger Activation parameter. The available settings for the Trigger Activation parameter are:

- Rising Edge - specifies that a rising edge of the source signal will start a line acquisition.
- Falling Edge - specifies that a falling edge of the source signal will start a line acquisition.

Typically, a frame grabber is used to supply an electrical line start signal to CC1, CC2, CC3, or CC4.

By default, CC1 is selected as the source signal for the line start trigger.

If the Line Start Trigger Source parameter is set to CC1, CC2, CC 3, or CC4, the externally generated electrical signal applied to the selected input line must be held high for a minimum period for the camera to detect a transition from low to high and must be held low for a minimum period for the camera to detect a transition from high to low:

- The minimum period is 100 ns if the debouncer is not used.
- The minimum period is equal to or larger than the debouncer value if the debouncer is used.

For more information about using a software trigger to control line acquisition start, see Section 6.1.4 on page 48.

For more information about using a hardware trigger to control line acquisition start, see Section 6.1.5 on page 52.

ⓘ	In all cases, the exposure time for each line must be within the minimum and the maximum stated in Table 7 on page 53. This is true regardless of the method used to control exposure.

Exposure Time Control with Trigger Mode = On

When the line start trigger mode is set to on, the exposure time for each line acquisition is determined by the value of the camera's exposure time parameters.

For more information about setting exposure time, see Section 6.1.3 on page 46.

6.1.3 Setting the Trigger Mode and Related Parameters

Setting the Parameters Using Basler pylon

You can set the Trigger Mode and related parameter values from within your application software by using the pylon API. If your settings make it necessary, you can also set the Trigger Source parameter.

The following code snippet illustrates using the API to set the Trigger Mode parameter for the line start trigger to on and the Trigger Source parameter to CC1:

```
// Select the line start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_LineStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_CC1 );
```

The following code snippet illustrates using the API to set the Trigger Mode parameter for the line start trigger to off and the Acquisition Line Rate Abs parameter to 60:

```
// Select the line start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_LineStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_Off );
// Set the exposure time in (µs)
Camera.ExposureTimeAbs.SetValue( 300 );
// Enable the acquisition line rate parameter and set the line rate. (Enabling
// the acquisition line rate parameter allows the camera to control the line
// rate internally.
Camera.AcquisitionLineRateEnable.SetValue( true );
Camera.AcquisitionLineRateAbs.SetValue( 60.0 );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting the Parameters Using Direct Register Access

To set the trigger mode to on and to select a trigger source via direct register access:

- Set the value of the Trigger Mode Line Start register to On.
- Set the value of the Trigger Source Line Start register to Software, CC1, CC2, CC3, or CC4.
- If the trigger source is set to CC1, CC2, CC3, or CC4, set the value of the Trigger Activation Line Start register to Rising Edge or Falling Edge.

To set the line start trigger mode to off, set the exposure time, and set the line acquisition rate via direct register access:

- Set the value of the Trigger Mode Line Start register to Off.
- Set the value of the Exposure Time Raw register as desired.

  A value in a raw register is simply an integer value with no units. To determine what the actual setting will be, you must multiply the value in the raw register by the camera's time base. The time base on racer cameras is 100 ns.

  For example, if you set the Exposure Time Raw register to 1000, the exposure time would be 100 µs (1000 x 100 ns = 100 µs).
- Set the value of the Acquisition Line Period Enable register to 1 (true).
  (This will enable the camera's ability to internally control the line period.)
- Set the value of the Acquisition Line Period Raw register as desired.
  (Line Rate = 1 /Line Period.)

For more information about direct register access, see Section 3.2 on page 16.

6.1.4 Using a Software Line Start Trigger Signal

6.1.4.1 Introduction

If the Line Start Trigger Mode parameter is set to on and the Trigger Source parameter is set to software, you must apply a software line start trigger signal to the camera to begin each line acquisition. Assuming that the camera is in a "waiting for line start trigger" acquisition status, line exposure will start when the software line start trigger signal is received by the camera. Fig. 10 illustrates line acquisition with a software line start trigger signal.

When the camera receives a software trigger signal and begins exposure, it will exit the "waiting for line start trigger" acquisition status because at that point, it cannot react to a new line start trigger signal. As soon as the camera is capable of reacting to a new line start trigger signal, it will automatically return to the "waiting for line start trigger" acquisition status.

Timed Exposure Mode

When you are using a software trigger signal to start each line acquisition, the camera’s Exposure Mode must be set to timed. The exposure time for each acquired line will be determined by the value of the camera’s Exposure Time Abs parameter if you are parameterizing the camera with Basler pylon or by the Exposure Time Raw register if you are parameterizing the camera via direct register access.

The minimum and the maximum allowed exposure time for each acquired line are as shown in Table 6.

	raL2048-80km	**raL4096-80km**
Min	2.0 µs	2.0 µs
Max	10000 µs	10000 µs

Table 6: Minimum and Maximum Allowed Exposure Times

ⓘ If you are using a GenICam compliant tool such as the Basler pylon Viewer and you attempt to set the exposure time to exactly the minimum allowed or to exactly the maximum allowed, you will see unusual error codes. This is an artifact of a rounding error in the GenICam interface architecture. As a work around, you could set the exposure time slightly above the minimum or below the maximum. Values between the minimum and the maximum are not affected by the problem.

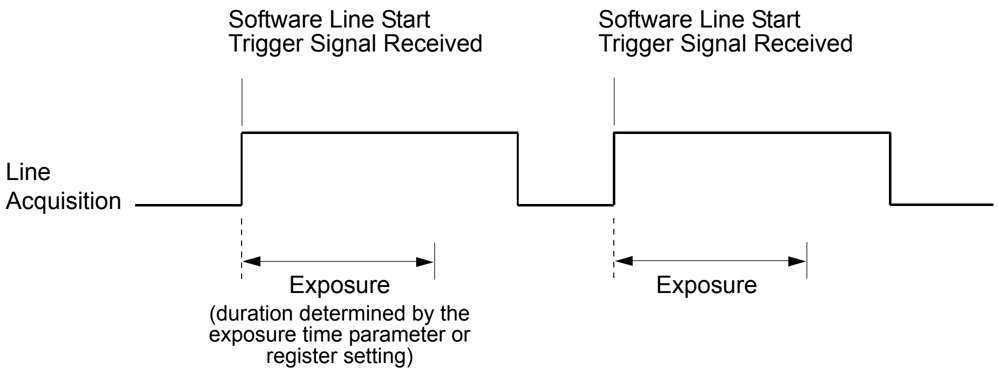


Fig. 10: Line Acquisition with a Software Line Start Trigger and Exposure Mode Set to Timed

When you are using a software trigger signal to start each line acquisition, the line rate will be determined by how often you apply a software trigger signal to the camera, and you should not attempt to trigger line acquisition at a rate that exceeds the maximum allowed with the current camera settings. (There is a detailed explanation about the maximum allowed line rate at the end of this chapter.) Software line start trigger signals that are applied to the camera when it is not ready to receive them will be ignored.

Section 6.1.4.2 includes more detailed information about applying a software line start trigger signal to the camera using Basler pylon or via direct register access.

For more information about determining the maximum allowed line rate with the current camera settings, see Section 6.4 on page 66.

6.1.4.2 Setting the Parameters Related to Software Line Start Triggering and Applying a Software Trigger Signal

Setting the Parameters and Applying the Signal Using Basler pylon

You can set all of the parameters needed to perform software line start triggering from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values and execute the commands related to software line start triggering.

```
// Disable the acquisition line rate parameter (this will disable the camera’s
// internal line rate control and allow you to control the line rate with
// software line start trigger signals)
Camera.AcquisitionLineRateEnable.SetValue( false );
// Select the line start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_LineStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Software );
```

```
// Set for the timed exposure mode
Camera.ExposureMode.SetValue( ExposureMode_Timed );
// Set the exposure time
Camera.ExposureTimeAbs.SetValue( 300 );

     while ( ! finished )
     {
         // Execute a trigger software command to apply a line start
         // trigger signal to the camera
         Camera.TriggerSoftware.Execute( );
         // Retrieve acquired line here
     }

// Note: as long as the Trigger Selector is set to LineStart, executing
// a Trigger Software command will apply a software line start trigger
// signal to the camera
```

The following code snippet illustrates using the API to check the acquisition status:

```
// Set the acquisition status selector
Camera.AcquisitionStatusSelector.SetValue
( AcquisitionStatusSelector_LineTriggerWait );

// Read the acquisition status
bool IsWaitingForLineTrigger = Camera.AcquisitionStatus.GetValue();
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting the Parameters and Applying the Signal Using Direct Register Access

To set the parameters needed to perform software line start triggering via direct register access:

- Set the value of the Acquisition Line Period Enable register to 0 (false).
  (This will disable the camera’s ability to internally control the line period and allow you to control the line rate with software trigger signals.)
- Set the value of the Trigger Mode Line Start register to On.
- Set the value of the Trigger Source Line Start register to Software.
- Set the value of the Exposure Mode register to Timed.
- Set the value of the Exposure Time Raw parameter as desired.

  A value in a raw register is simply an integer value with no units. To determine what the actual setting will be, you must multiply the value in the raw register by the camera’s time base. The time base on racer cameras is 100 ns.

  For example, if you set the Exposure Time Raw register to 1000, the exposure time would be 100 µs (1000 x 100 ns = 100 µs).
- Set the value of the Trigger Software Line Start register to 1.
  Setting the value of this register to 1 applies a software line start trigger to the camera. The register resets to 0 when execution is complete.

To determine the acquisition status of the camera via direct register access:

- Read the value of the Status Line Trigger Wait register.
  A value of 0 indicates that the camera is not ready to receive a line start trigger.
  A value of 1 indicates that the camera is ready to receive a line start trigger.

For more information about direct register access, see Section 3.2 on page 16.

6.1.5 Using a Hardware Line Start Trigger Signal

6.1.5.1 Introduction

If the Trigger Mode parameter for the line start trigger is set to on and the Trigger Source parameter is set to CC1, CC2, CC3, or CC4, an externally generated electrical signal injected into the selected source will act as the line start trigger signal for the camera. This type of trigger signal is generally referred to as a hardware trigger signal or as an external line start trigger signal (ExLSTrig signal).

A rising edge or a falling edge of the ExLSTrig signal can be used to trigger line acquisition. The Trigger Activation parameter is used to select rising edge or falling edge triggering.

Assuming that the camera is in a "waiting for line start trigger" acquisition status, line acquisition will start when the hardware line start trigger signal is received by the camera.

When the camera receives a hardware trigger signal and begins exposure, it will exit the "waiting for line start trigger" acquisition status because at that point, it cannot react to a new line start trigger signal. As soon as the camera is capable of reacting to a new line start trigger signal, it will automatically return to the "waiting for line start trigger" acquisition status.

When the camera is operating under control of an ExLSTrig signal, the period of the ExLSTrig signal will determine the rate at which the camera will acquire lines:

$$\frac{1}{\text{ExLSTrig period in seconds}} = \text{Line Rate}$$

For example, if you are operating a camera with an ExLSTrig signal period of 20 ms (0.020 s):

$$\frac{1}{0.020} = 50\text{ lps}$$

So in this case, the line rate is 50 fps.

If you have selected CC1, CC2, CC3, or CC4 as the trigger source, your frame grabber will typically apply the electrical signal to the selected input via the Camera Link cable. For more information about applying an ExLSTrig signal to CC1, CC2, CC3, or CC4, see the documentation for your frame grabber.

ⓘ

If you are triggering line acquisition with an ExLSTrig signal and you attempt to acquire lines at too high a rate, some of the line trigger signals that you apply will be received by the camera when it is not in a "waiting for line start trigger" acquisition status. The camera will ignore any line start trigger signals that it receives when it is not "waiting for line start trigger". (This situation is commonly referred to as "overtriggering" the camera.

To avoid overtriggering, you should not attempt to acquire lines at a rate that exceeds the maximum allowed with the current camera settings.

For more information about determining the maximum allowed line rate with the current camera settings, see Section 6.4 on page 66.

For more information about setting the camera for hardware triggering and selecting the source to receive the ExLSTrig signal, see Section 6.1.5 on page 52.

For more information about CC1, CC2, CC3, and CC4, see Section 5.6.1 on page 31.

6.1.5.2 Exposure Modes

If you are triggering the start of line acquisition with an externally generated line start trigger (ExLSTrig) signal, the following exposure modes are available:

- Timed exposure mode
- Trigger width exposure mode

Timed Exposure Mode

When the timed exposure mode is selected, the exposure time for each line acquisition is determined by:

- the value of the camera’s Exposure Time Abs parameter if you are parameterizing the camera with Basler pylon.
- the value of the Exposure Time Raw register if you are parameterizing the camera via direct register access.

The minimum and the maximum allowed exposure time for each acquired line are as shown in Table 6.

	raL2048-80km	**raL4096-80km**
Min	2.0 µs	2.0 µs
Max	10000 µs	10000 µs

Table 7: Minimum and Maximum Allowed Exposure Times

ⓘ If you are using a GenICam compliant tool such as the Basler pylon Viewer and you attempt to set the exposure time to exactly the minimum allowed or to exactly the maximum allowed, you will see unusual error codes. This is an artifact of a rounding error in the GenICam interface architecture. As a work around, you could set the exposure time slightly above the minimum or below the maximum. Values between the minimum and the maximum are not affected by the problem.

If the camera is set for rising edge triggering, the exposure time starts when the ExLSTrig signal rises. If the camera is set for falling edge triggering, the exposure time starts when the ExLSTrig signal falls. Fig. 11 illustrates timed exposure with the camera set for rising edge triggering.

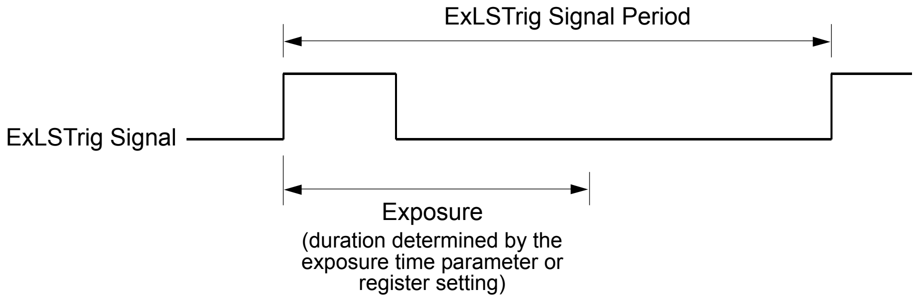

ExLSTrig Signal Period
ExLSTrig Signal
Exposure
(duration determined by the exposure time parameter or register setting)

Fig. 11: Timed Exposure with Rising Edge Triggering

Note that if you attempt to trigger a new exposure start while the previous exposure is still in progress (overtriggering), the trigger signal will be ignored. This situation is illustrated in Fig. 12 for rising edge triggering.

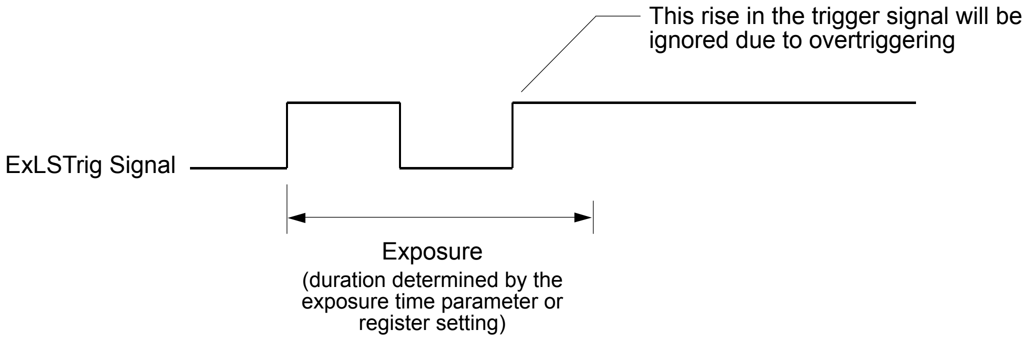

This rise in the trigger signal will be ignored due to overtriggering
ExLSTrig Signal
Exposure
(duration determined by the exposure time parameter or register setting)

Fig. 12: Overtriggering with Timed Exposure

Trigger Width Exposure Mode

When trigger width exposure mode is selected, the length of the exposure for each line acquisition will be directly controlled by the ExLSTrig signal. If the camera is set for rising edge triggering, the exposure time begins when the ExLSTrig signal rises and continues until the ExLSTrig signal falls. If the camera is set for falling edge triggering, the exposure time begins when the ExLSTrig signal falls and continues until the ExLSTrig signal rises.

The minimum allowed exposure time for each acquired line is as shown in Table 8 and there is no limit on the maximum exposure time. Keep in mind, however, that using a very long exposure time can lead to significant degradation of the image quality.

	raL2048-80km	raL4096-80km
Min	2.0 µs	2.0 µs

Table 8: Minimum Allowed Exposure Times

If you are using a GenICam compliant tool such as the Basler pylon Viewer and you attempt to set the exposure time to exactly the minimum allowed or to exactly the maximum allowed, you will see unusual error codes. This is an artifact of a rounding error in the GenICam interface architecture. As a work around, you could set the exposure time slightly above the minimum or below the maximum. Values between the minimum and the maximum are not affected by the problem.

Fig. 13 illustrates trigger width exposure with the camera set for rising edge triggering.

Trigger width exposure is especially useful if you intend to vary the length of the exposure time for each captured line.

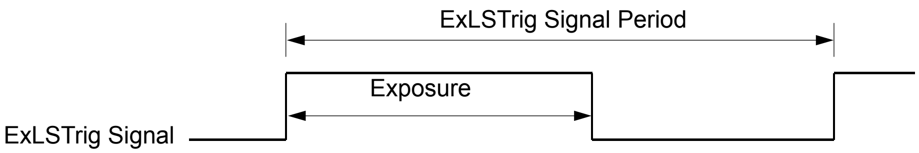


Fig. 13: Trigger Width Exposure with Rising Edge Triggering

When you operate the camera in trigger width exposure mode, you must also set the camera's Exposure Overlap Time Max Abs parameter if you are parameterizing the camera with Basler pylon. If you are parameterizing the camera via direct register access you must set the value of the Exposure Overlap Time Max register. The Exposure Overlap Time Max setting will be used by the camera to operate the Line Trigger Wait signal.

You should set Exposure Overlap Time Max to represent the shortest exposure time you intend to use. For example, assume that you will be using trigger width exposure mode and that you intend to use the ExLSTrig signal to vary the exposure time in a range from 3000 µs to 5500 µs. In this case you would set the camera’s Exposure Overlap Time Max to 3000 µs.

For more information about the Line Trigger Wait signal, see Section 6.3.3 on page 64

6.1.5.3 Setting the Parameters Related to Hardware Line Start Triggering and Applying a Hardware Trigger Signal

Setting the Parameters Using Basler pylon and Applying the Signal

You can set all of the parameters needed to perform hardware line start triggering from within your application software by using the pylon API.

The following code snippet illustrates using the API to set the parameter values and execute the commands related to hardware line start triggering with the camera set for the timed exposure mode with rising edge triggering and CC1 as the trigger source.

```
// Disable the acquisition line rate parameter (this will disable the camera’s
// internal line rate control and allow you to control the line rate with
// external line start trigger signals)
Camera.AcquisitionLineRateEnable.SetValue( false );
// Select the line start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_LineStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_CC1 );
// Set the trigger activation mode to rising edge
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
// Set for the timed exposure mode
Camera.ExposureMode.SetValue( ExposureMode_Timed );
// Set the exposure time
Camera.ExposureTimeAbs.SetValue( 300 );

// Line acquisition will start each time the externally generated
// line start trigger signal (ExLSTrig signal)goes high
```

The following code snippet illustrates using the API to set the parameter values and execute the commands related to hardware line start triggering with the camera set for the trigger width exposure mode with rising edge triggering and with line CC1 as the trigger source.

```
// Disable the acquisition line rate parameter (this will disable the camera’s
// internal line rate control and allow you to control the line rate with
// external line start trigger signals)
Camera.AcquisitionLineRateEnable.SetValue( false );
// Select the line start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_LineStart );
```

```
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger to CC1
Camera.TriggerSource.SetValue ( TriggerSource_CC1 );
// Set the trigger activation mode to rising edge
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
// Set for the trigger width exposure mode
Camera.ExposureMode.SetValue( ExposureMode_TriggerWidth );

// Line acquisition will start each time the externally generated
// line start trigger signal (ExLSTrig signal)goes high
```

You can also use the Basler pylon Viewer application to easily set the parameters.

Setting the Parameters Using Direct Register Access and Applying the Signal

To set the parameters needed to perform hardware line start triggering via direct register access:

- Set the value of the Acquisition Line Period Enable register to 0 (false).
  (This will disable the camera’s ability to control the line period internally and allow you to control the line rate with an external signal.)
- Set the value of the Trigger Mode Line Start register to On.
- Set the value of the Trigger Source Line Start register to receive the external trigger signal on, CC1, CC2, CC3, or CC4.
- Set the value of the Trigger Activation Line Start register to Rising Edge or Falling Edge as desired.
- Set the value of the Exposure Mode register to Timed or to Trigger Width.

  If the mode is set to timed, set the value of the Exposure Time Raw register as desired.

  A value in a raw register is simply an integer value with no units. To determine what the actual setting will be, you must multiply the value in the raw register by the camera’s time base. The time base on racer cameras is 100 ns.

  For example, if you set the Exposure Time Raw register to 1000, the exposure time would be 100 µs (1000 x 100 ns = 100 µs).

Apply the appropriate externally generated electrical signal (ExLSTrig signal) to the selected trigger source.

For more information about direct register access, see Section 3.2 on page 16.

Exposure Start and Exposure End Delays

When the line start trigger mode is set to on and an input line is selected as the source signal for the line start trigger, there is a delay between the transition of the line start signal and the actual start of exposure. For example, if you are using the timed exposure mode with rising edge triggering, there is a delay between the rise of the signal and the actual start of exposure.

There is also an exposure end delay, i.e., a delay between the point when exposure should end as explained in the diagrams on the previous page and when it actually does end.

The base exposure start and end delays are as shown in Table 9:

	raL2048-80km	raL4096-80km
Start Delay	1.5 µs	1.5 µs
End Delay	1.2 µs	1.2 µs

Table 9: Base Exposure Start and End Delays

ⓘ If you are using a GenICam compliant tool such as the Basler pylon Viewer and you attempt to set the exposure time to exactly the minimum allowed or to exactly the maximum allowed, you will see unusual error codes. This is an artifact of a rounding error in the GenICam interface architecture. As a work around, you could set the exposure time slightly above the minimum or below the maximum. Values between the minimum and the maximum are not affected by the problem.

ⓘ When using the frequency converter, the delay values may slightly differ from those given in Table 9.

There is also a second component to the start and end delays. This second component is the debouncer setting for the input line. The debouncer setting for the input line must be added to the base start and end delays shown in Table 9 to determine the total start delay and end delay.
For example, assume that you are using an ruL2048-48gm camera and that you have set the line start trigger mode to on. Also assume that you have selected input line CC1 as the source signal for the line start trigger and that the debouncer parameter for line CC1 is set to 5 µs. In this case:

Total Start Delay = Start Delay Value from Table 9 + Debouncer Setting

Total Start Delay = 1.5 µs + 5 µs

Total Start Delay = 6.5 µs

Total End Delay = End Delay Value from Table 9 + Debouncer Setting

Total End Delay = 1.2 µs + 5 µs

Total End Delay = 6.2 µs

6.2 Frequency Converter

The camera is equipped with a frequency converter module that allows triggering the camera at a frequency that differs from the frequency of the input signals received.

The frequency converter module includes three sub-modules acting in sequence on the original signals:

- The pre-divider module receives the input signals. The module allows employing an integer factor, the pre-divider, to decrease the original frequencies and passes the signals on to the next module, the multiplier module.

  If for example a pre-divider of 2 is selected only every other input signal is passed out unchanged to the multiplier module and, accordingly, the frequency is halved. If a pre-divider of 1 is selected every input signal is passed out unchanged to the multiplier module.

  Employing the pre-divider may be advisable for decreasing periodic jitter of the input signals and will be required if the input signal frequency is higher than 100 kHz. The signal frequency of the signals passed on to the multiplier module must be within the range of 10 Hz to 100 kHz.

  We recommend to only use low values for the pre-divider. The original signal frequency should be changed as little as possible to facilitate frequency adjustment by the multiplier module.

- The multiplier module receives the signals from the pre-divider module. The signal frequency must be within the range of 10 Hz to 100 kHz. The multiplier module allows applying an integer factor, the multiplier, to generate signals at increased frequencies and passes the signals on to the next module, the post-divider module.

  If, for example, a multiplier of 2 is selected signals are generated at double the frequency of the signals received from the pre-divider module and are passed on to the divider module. If a multiplier of 1 is selected every signal received from the pre-divider module is passed unchanged on to the divider module.

  The Align parameter can be set to "rising edge" and "falling edge". If "rising edge" is selected there will be for the rising edge of each signal received from the pre-divider module a phase-locked, matching rising edge among the signals generated. If "falling edge" is selected there will be for the falling edge of each signal received from the pre-divider module a phase-locked, matching falling edge among the signals generated.

  Make sure to select a multiplier that will not too much increase the frequency such that the camera will be overtriggered. Temporarily, a too high frequency may occur during frequency adjustment causing overtriggering even if a relatively low multiplier was selected. We recommend to use one of the safeguards against overtriggering the camera:

  - Set the PreventOvertrigger parameter to True if you use Basler pylon for camera configuration
  - Set the Frequency Converter Prevent Overtrigger register to 1 if you use direct register access for camera configuration.

  The PreventOvertrigger parameter provides a safeguard against overtriggering the camera. We recommend setting the PreventOvertrigger parameter to True to prevent overtriggering.

- The post-divider module receives the signals from the multiplier module. The post-divider module allows employing an integer factor, the post-divider, to generate signals at decreased frequencies and provides these signals to be used as camera trigger signals, e.g. as line start triggers.

  If for example a post-divider of 2 is selected only every other signal received from the multiplier module is passed out from the divider module and, accordingly, the frequency is halved. If a post-divider of 1 is selected every signal received from the multiplier module is passed out unchanged from the divider module.

> You can use the frequency converter to multiply the original signal frequency by a fractional value. We recommend multiplying the frequency by the enumerator value using the multiplier module and dividing the resulting frequency by the denominator value using the post-divider module.

Setting the Parameters Using Basler pylon

You can configure the frequency converter module from within your application by using a dynamic API. The following code snippet illustrates setting parameter values:

```
INodeMap &Control = *Camera.GetNodeMap();

// possible values for FrequencyConverterInputSource:
// CC1
// CC2
// CC3
// CC4
CEnumerationPtr(Control.GetNode("FrequencyConverterInputSource"))-
>FromString("CC1");

// ranges for divider and multiplier:
// divider   : 1...128
// multiplier: 1...32
CIntegerPtr(Control.GetNode("FrequencyConverterPreDivider"))->SetValue(4);
CIntegerPtr(Control.GetNode("FrequencyConverterMultiplier"))->SetValue(17);
CIntegerPtr(Control.GetNode("FrequencyConverterPostDivider"))->SetValue(1);
```

You can also use the Basler pylon Viewer application to easily set the parameters.

Setting the Parameters Using Direct Register Access

To set the parameters needed to configure the frequency converter module via direct register access:

- Select the source signal for frequency converstion by setting the related value of the Frequency Converter Signal Source register.
- Set the desired frequency value of the Frequency Converter Pre Divider register.
- If you want to perform signal alignment (see above) you can set the value of the Frequency Converter Signal Alignment register.
- To avoid overtriggering, we recommend to set the Frequency Converter Prevent Overtrigger register to 1.

- Set the desired frequency value of the Frequency Converter Multiplier register.
- Set the desired frequency value of the Frequency Converter Post Divider register.

For more information about direct register access, see Section 3.2 on page 16.

6.3 Acquisition Monitoring Tools

The camera includes the acquisition status feature and generates these output signals that you can use to monitor the progress of line acquisition by the camera:

- the exposure active signal
- the line trigger wait signal.

The camera also allows selecting the output of the frequency converter module as output signals.

In addition, you can check the camera's acquisition status (see below).

6.3.1 Exposure Active Signal

The camera's Exposure Active output signal will go high when the exposure time for each line acquisition begins and goes low when the exposure time ends. The signal can for example be used as a flash trigger.

An example of the Exposure Active signal's behavior on a camera using a rising edge external line start trigger signal (ExLSTrig) and the timed exposure mode is shown in Fig. 14.

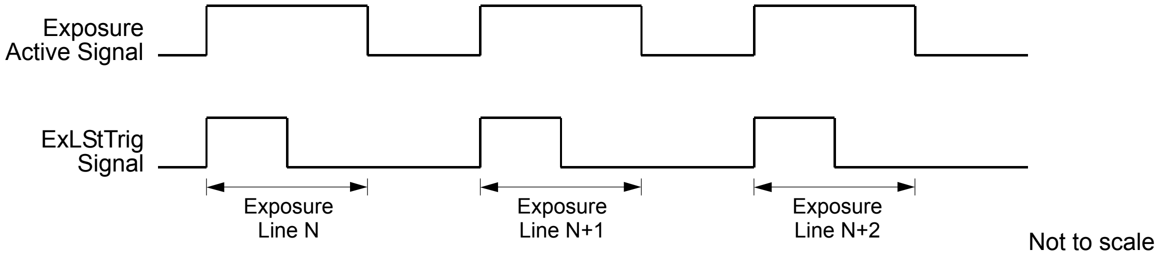


Fig. 14: Exposure Active Signal

Selecting the Exposure Active Signal as the Source Signal for the CLSpare Output Line Using Basler pylon

You can select the exposure active signal as the source signal for the CL Spare output line:

- Use the Line Selector to select the CLSpare line.
- Set the value of the Line Source Parameter to the exposure active output signal.

You can set the Line Selector and the Line Source parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
//Select the CL Spare line
Camera.LineSelector.SetValue( LineSelector_ClSpare );
//Set the source signal for the selected line
Camera.LineSource.SetValue( LineSource_ExposureActive );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

Selecting the Exposure Active Signal as the Source Signal for the CLSpare Output Line Using Direct Register Access

You can select the exposure active signal as the source signal for the camera’s CL Spare output line.

To select the exposure active signal as the source signal for the CL Spare line via direct register access:

- Set the value of the Line Source CL Spare register to Exposure Active.

For more information about direct register access, see Section 3.2 on page 16.

6.3.2 Acquisition Status

If a camera receives a software line start trigger signal when it is not in a "waiting for line start trigger" acquisition status, it will simply ignore the trigger signal.

The camera’s acquisition status indicator gives you the ability to check whether the camera is in a "waiting for line start trigger" acquisition status. If you check the acquisition status before you apply each software line start trigger signal, you can avoid applying trigger signals to the camera that will be ignored.

The acquisition status indicator is designed for use when you are using host control of image acquisition, i.e., when you are using software line start trigger signals.

To determine the acquisition status of the camera via the Basler pylon API:

- Use the Acquisition Status Selector to select the Line Trigger Wait status.
- Read the value of the Acquisition Status parameter.

  If the value is set to "false", the camera is not waiting for the trigger signal.

  If the value is set to "true", the camera is waiting for the trigger signal.

Checking the Acquisition Status Using Basler pylon

To determine the acquisition status of the camera via the Basler pylon API:

- Use the Acquisition Status Selector to select the Line Trigger Wait status.
- Read the value of the Acquisition Status parameter.

  If the value is set to "false", the camera is not waiting for the trigger signal.

  If the value is set to "true", the camera is waiting for the trigger signal.

You can check the acquisition status from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to check the acquisition status:

```
// Check the line start trigger acquisition status
// Set the acquisition status selector
Camera.AcquisitionStatusSelector.SetValue
( AcquisitionStatusSelector_LineTriggerWait );
// Read the acquisition status
bool IsWaitingForLineTrigger = Camera.AcquisitionStatus.GetValue();
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Checking the Acquisition Status Using Direct Register Access

To determine the line start trigger status via the direct register access:

- Read the value of the Status Line Trigger Wait register.

  If the value is set to 0, the camera is not waiting for the trigger signal.

  If the value is set to 1, the camera is waiting for the trigger signal.

For more information about direct register access, see Section 3.2 on page 16.

6.3.3 Line Trigger Wait Signal

The camera's Line Trigger Wait output signal will be low when the camera is in the process of acquiring a line and is not able to accept a new line start trigger. As soon as the current line acquisition is complete and the camera is ready to acquire a new line, the Line Trigger Wait signal will go high.

This signal can be selected as the source signal for the CLSpare output line on the camera.

Selecting the Line Trigger Wait Signal as the Source Signal for the CLSpare Output Line Using Basler pylon

To select the line trigger wait signal as the source signal for the CL Spare line using Basler pylon:

- Select the line trigger wait signal as the source signal for the camera's CL Spare output line: Use the Line Selector to select the CL Spare output line.
- Set the value of the Line Source Parameter to the line trigger wait signal.

You can set the Line Selector and the Line Source parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
//Select the CL Spare line
Camera.LineSelector.SetValue( LineSelector_ClSpare );
//Set the source for the selected line
Camera.LineSource.SetValue( LineSource_LineTriggerWait );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1.1 on page 16.

Selecting the Line Trigger Wait Signal as the Source Signal for the CLSpare Output Line Using Direct Register Access

To select the line trigger wait signal as the source signal for the CL Spare line via direct register access:

- Set the value of the Line Source CL Spare register to Line Trigger Wait.

For more information about direct register access, see Section 3.2 on page 16.

6.3.4 Input Related Signals as Output Signals

The camera allows selecting the output signal of the frequency converter module and assigning it to the camera's CLSpare output line. In this fashion input signals can be passed through a camera to trigger additional cameras.

In this case, setting a minimum output pulse width may be necessary to ensure output signal detection.

For more information about selecting the source signal for the CLSpare output line on the camera, see Section 5.6.2.3 on page 37.

For more information about the minimum output pulse width feature, see Section 5.6.2.1 on page 36.

6.4 Maximum Allowed Line Acquisition Rate

The maximum allowed line acquisition rate for your camera is not static. It can vary depending on how certain camera features are set. In general, the following factors can affect the maximum allowed line rate:

- The Camera Link pixel clock speed and the Camera Link tap geometry settings.

  When the pixel clock speed is set to a high value, it will take less time to transfer captured images from the camera to the frame grabber in your host PC.

  When the camera is set for a tap geometry that uses more taps (e.g., the 1X4 geometry uses four taps and the 1X8 geometry uses eight taps), it can typically transfer data out of the camera faster.

  So if the camera is set for a high pixel clock speed and a high number of taps, it typically will have a much higher maximum allowed line rate than when it is set for a low pixel clock speed and a low number of taps.
- Whether binning is enabled. Enabling horizontal binning may yield a significant increase in the maximum allowed line rate, but this will only be true if the camera is not already reading out image data at a rate that is near to the current data carrying capacity of the Camera Link interface.
- The exposure time for acquired lines. If you use very long exposure times, you can acquire fewer lines per second.

There are several ways that you can determine the maximum allowed acquisition line rate with your current camera settings:

- You can go to the Support section of the Basler website and use the online frame rate calculator:

  www.baslerweb.com
- You can use Basler pylon to read the value of the camera’s Resulting Line Rate Abs parameter (see below).
- You can use direct register access to read the value of the Resulting Line Period Raw register (see below).

For more information about selectable Camera Link pixel clock speeds, see Section 8.1 on page 75.

For more information about Camera Link tap geometries, see Section 7.2 on page 73.

Using Basler pylon to Check the Maximum Allowed Line Rate

You can use the Basler pylon API to read the current value of the Resulting Line Rate Abs parameter from within your application software using the Basler pylon API. The following code snippet illustrates using the API to get the parameter value:

```
// Get the resulting line rate
double resultingLps = Camera.ResultingLineRateAbs.GetValue();
```

The Resulting Line Rate Abs parameter takes all camera settings that can influence the line rate into account and indicates the maximum allowed line rate given the current settings.

You can also use the Basler pylon Viewer application to easily read the parameter.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Using Direct Register Access to Check the Maximum Allowed Rate

When using direct register access, you work with the "resulting line period" rather than the line acquisition rate. Once the resulting line period is known, the maximum allowed line acquisition rate can be determined by:

$$\text{MaxLine Acquisition Rate} = \frac{1}{\text{Resulting Line Period in Seconds}}$$

To determine the resulting line period:

- Check the value of the Resulting Line Period Raw register.

The Resulting Line Period Raw register takes all of camera settings that can influence the line period into account and indicates the minimum allowed line period given the current settings.

A value in a raw register is simply an integer value with no units. To determine what the actual line period will be, you must multiply the value in the raw register by the camera's time base. The time base on racer cameras is 1 µs.

For example, if you read the Resulting Line Period Raw register and find that its value is 10000, the resulting line period would be 10000 µs (10000 x 1 µs = 10000 µs).

For more information about direct register access, see Section 3.2 on page 19.

6.4.1 Increasing the Maximum Allowed Line Rate

You may find that you would like to acquire lines at a rate higher than the maximum allowed with the camera's current settings. In this case, you must first determine what factor is most restricting the maximum line rate. The descriptions of the three factors that appear below will let you determine which factor is restricting the rate.

Factor 1:

Factor 1 is the sensor readout time. The readout time for a particular sensor is a fixed value and thus the maximum line acquisition rate as determined by the readout time is also fixed. The table below shows the maximum line rate (in lines per second) based on sensor readout time for each camera model.

Max Lines/s (based on sensor readout)	
raL2048-80km	**raL4096-80km**
80000	80000

Factor 2:

Factor 2 is the exposure time. You can use the formula below to calculate the maximum line rate based on the exposure time for each acquired line:

$$\text{Max Lines/s} = \frac{1}{\text{Exposure time in µs} + C_1}$$

Where the constant C_1 depends on the camera model as shown in the table below:

	raL2048-80km	**raL4096-80km**
C_1	3.1 µs	3.1 µs

For more information about setting the exposure time, see Section 6.1.5.3 on page 56.

Formula 3 calculates the maximum line rate based on the amount of time it takes to transmit the pixel data for an acquired line from the camera to the host PC:

$$\text{Max Lines / s} = \frac{\text{CL Clk} \times \text{Taps}}{(9 \times \text{Taps}) + \text{AOI Width}}$$

Where: *CL Clk* is the Camera Link clock speed for your camera model (see Section 8.1 on page 75

Taps is the number of Camera Link taps being used as determined by the current Camera Link tap geometry setting (see Section 7.2 on page 73)

AOI Width is the width of the AOI based on the current AOI width setting (see Section 8.4 on page 83)

Once you have determined the maximum allowed line rate, you can easily determine the minimum allowed line period:

$$\text{Min Line Period} = \frac{1}{\text{Max Line Rate}}$$

Example

Assume that you are using an raL2048-80km camera set for an exposure time of 190 µs. Also assume that you are using full resolution (2048 pixels), a Camera Link clock speed of 32.5 MHz, and the 1X4 Camera Link tap geometry that provides 4 taps.

Factor 1 (sensor readout):

Max Lines/s = 80000

Factor 2 (exposure time):

$$\text{Max Lines/s} = \frac{1}{190\ \mu s + 3.1\ \mu s}$$

Max Lines/s = 5178 Lines/s

Formula 3 (data transmission):

$$\text{Max Lines / s} = \frac{32500000 \times 4}{(9 \times 4) + 2048}$$

Max Lines/s = 62380

Factor 2, the exposure time, is the most restrictive factor. In this case, the exposure time setting is limiting the maximum allowed line rate to 5178 lines per second. If you wanted to operate the camera at a higher line rate, you would need to lower the exposure time.

7 Pixel Formats and Tap Geometries

This chapter provides information about the sensor bit depths, pixel formats, and pixel clock speeds available on the camera.

ⓘ For a description of tap geometries and complete details regarding the way that pixel data is handled by the camera, refer to the Basler document named *racer Camera Link Information for Frame Grabber Designers* (AW001187xx000). You can obtain the document from the camera's Downloads section of the Basler website: www.baslerweb.com

7.1 Pixel Formats

Pixel Formats for Monochrome Cameras

The choice of a pixel format determines the bit depth of the data transmitted from the camera for each pixel in the acquired lines. The pixel formats available on the mono cameras depend on the currently selected Camera Link tap geometry as shown in Table 10.

Camera Link Tap Geometry	Available Pixel Formats		
	Mono 8	Mono 10	Mono 12
1X2	●	●	●
1X4	●	●	●
1X8	●	●	
1X10	●		

Table 10: Selected Camera Link Tap Geometry and Available Mono Pixel Formats (● = format available)

Pixel Format Interaction with the Sensor Bit Depth

The camera’s sensor can capture image data at 12 bit depth. There is an interaction between the Sensor Bit Depth setting and the Pixel Format setting as described in Table 11.

Sensor Bit Depth	**Pixel Format Setting**		
	Mono 8	**Mono 10**	**Mono 12**
12 bits	8 most significant bits of data from the sensor are transmitted	10 most significant bits of data from the sensor are transmitted	All 12 bits of data from the sensor are transmitted

Table 11: Interaction Between the Sensor Bit Depth and the Pixel Format

By selecting a pixel data format, you determine the format (layout) of the image data transmitted by the camera. This section provides detailed information about the available pixel data formats.

Setting the Pixel Format Using Basler pylon

You can use the pylon API to set the Pixel Format parameter value from within your application. The following code snippet illustrates using the API to set the parameter value:

```
// Set pixel format to Mono 8
Camera.PixelFormat.SetValue( PixelFormat_Mono8 );
// Set pixel format to Mono 10
Camera.PixelFormat.SetValue( PixelFormat_Mono10 );
// Set pixel format to Mono 12
Camera.PixelFormat.SetValue( PixelFormat_Mono12 );
```

You can also use the Basler pylon Viewer application to easily set the parameter.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 17.

Setting the Pixel Format Using Direct Register Access

To set the sensor pixel format via direct register access:

- Set the value of the Pixel Format register to Mono 8, Mono 10, or Mono 12 as desired.

For more information about direct register access, see Section 3.2 on page 19.

7.2 Camera Link Tap Geometry

The Camera Link tap geometry determines how the data that is read out of the imaging sensor will be transmitted from the camera to the frame grabber in your host PC via the Camera Link interface. The selection of a camera link tap geometry also determines whether your camera will be operating in the base, medium, or full Camera Link configuration and which Camera Link connectors on the camera will be used to transmit pixel data. Table 12 indicates how the Camera Link interface will operate with each available tap geometry setting.

The X2, X4, X8, or X10 in the tap geometry names indicates the number of Camera Link taps that will be used for a given configuration (e.g., X2 means 2 taps). As a general rule of thumb, a camera will have a higher maximum allowed line rate when it is operating with a tap geometry that uses more Camera Link taps.

Tap Geometry Setting	Camera Link Configuration	Camera Link Connectors Used to Transmit Data
1X2	Base	Base Only
1X4	Medium	Base and Medium/Full
1X8	Full	Base and Medium/Full
1X10	Full	Base and Medium/Full

Table 12: Camera Link Operation at Various Tap Geometry Settings

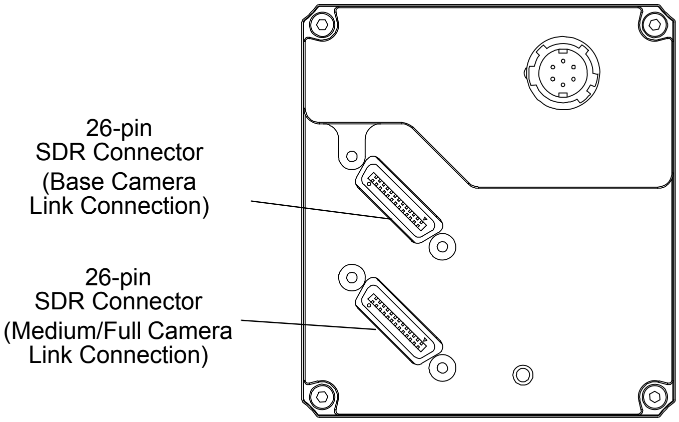


Fig. 15: Camera Link Connections

For a description of tap geometries refer to the Basler document named *racer Camera Link Information for Frame Grabber Designers* (AW001187xx000). You can obtain the document from the camera’s Downloads section of the Basler website: www.baslerweb.com

7.2.1 Setting the Tap Geometry

Setting the Tap Geometry Using Basler pylon

You can use the pylon API to set the Camera Link tap geometry from within your application software. The following code snippet illustrates using the API to set the tap geometry:

```
// Set the tap geometry to 1X2
Camera.ClTapGeometry.SetValue( ClTapGeometry_Geometry1X2 );

// Set the tap geometry to 1X4
Camera.ClTapGeometry.SetValue( ClTapGeometry_Geometry1X4 );

// Set the tap geometry to 1X8
Camera.ClTapGeometry.SetValue( ClTapGeometry_Geometry1X8 );

// Set the tap geometry to 1X10
Camera.ClTapGeometry.SetValue( ClTapGeometry_Geometry1X10 );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting the Tap Geometry Using Direct Register Access

To set the Camera Link tap geometry via direct register access:

- Set the value of the CL Tap Geometry register for 1X2, 1X4, 1X8 or 1X10 as desired.

For more information about direct register access, see Section 3.2 on page 16.

8 Features

This chapter provides detailed information about the features available on each camera. It also includes an explanation of their operation and the parameters associated with each feature.

8.1 Camera Link Pixel Clock Speed

The camera features selectable Camera Link pixel clock speeds. The pixel clock speed determines the rate at which pixel data will be transmitted from the camera to the frame grabber in your PC via the Camera Link interface. The available pixel clock speeds are: 32.5 MHz, 48 MHz, and 65 MHz, and 83.5 MHz. The default clock speed is 83.5 MHz.

Setting the camera for a higher pixel clock speed will increase the rate at which image data is transferred from the camera to the frame grabber. Some frame grabbers, however, cannot operate at the higher clock speeds. So it is important that you determine the maximum clock speed that your frame grabber can handle and that you set the camera’s speed no higher than the frame grabber’s maximum.

Keep in mind that if you set the camera for one of the lower pixel clock speeds, it may limit the camera’s maximum allowed line acquisition rate.

If you change the clock speed while the camera is in the process of acquiring images:

- Triggering and image acquisition will stop.
- Any acquired image that is being transmitted will be delivered.
- The camera’s clock speed will be changed internally.
- Once the change is complete, triggering and image acquisition will resume.

The Camera Link clock speed setting will be stored in the camera’s configuration sets. This means, for example, that if you have a different clock speed setting stored in user set 1 and user set 2 and you change the active set from user set 1 to user set 2, the clock speed will change.

For more information about the maximum allowed line acquisition rate, see Section 6.4 on page 66.

For more information about configuration sets, see Section 8.13 on page 108.

Setting the Camera Link Pixel Clock Using Basler pylon

You can use the pylon API to set the Camera Link pixel clock speed from within your application software. The following code snippet illustrates using the API to set the clock speed:

```
// Set the Camera Link pixel clock speed 32.5
Camera.ClPixelClock.SetValue( ClPixelClock_PixelClock32_5 );

// Set the Camera Link pixel clock speed to 48
```

```
    Camera.ClPixelClock.SetValue( ClPixelClock_PixelClock48 );
```

ⓘ	You can use the pylon API to set the pixel clock speed to 32.5, 48, 65, or 83.5 MHz. These are the only valid values for the pixel clock speed. If you attempt to use the API to set the clock speed to a value other then these, the camera will automatically round the setting down to the nearest valid speed.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting the Camera Link Pixel Clock Using Direct Register Access

To set the Camera Link pixel clock speed via direct register access:

- Set the value of the Cl Pixel Clock register for 32.5, 48, 65, or 83.5 MHz.

For more information about changing settings via direct register access, see Section 3.2 on page 16.

8.2 Gain and Black Level

8.2.1 Gain

The camera's gain is adjustable. As shown in Fig. 16, increasing the gain increases the slope of the response curve for the camera. This results in an increase in the gray values output from the camera for a given amount of output from the imaging sensor. Decreasing the gain decreases the slope of the response curve and results in lower gray values for a given amount of sensor output.

Increasing the gain is useful when at your brightest exposure, the highest gray values achieved are lower than 255 (for pixel data formats with 8 bit depth) or 4095 (for pixel data formats with 12 bit depth). For example, if you found that at your brightest exposure the gray values output by the camera were no higher than 127 (in an 8 bit format), you could increase the gain to 6 dB (an amplification factor of 2) and thus reach gray values of 254.

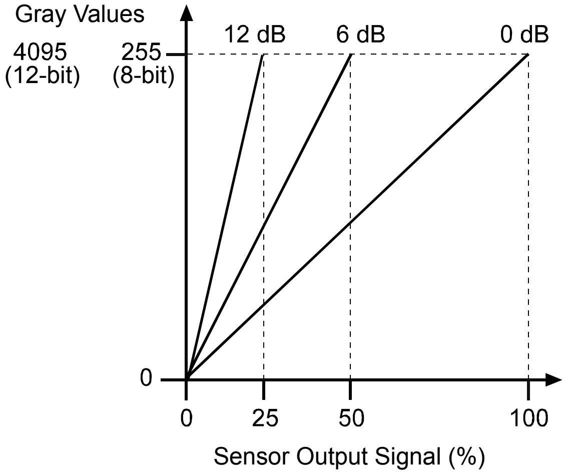


Fig. 16: Gain in dB

You can use the analog gain for coarsely setting gain and the digital gain for finer adjustment.

8.2.1.1 Analog Gain

The camera's analog gain is determined by the Gain parameter with the gain selector set to Analog All. All pixels in the sensor are affected by this setting.

The allowed parameter values are 1 and 4. A parameter value of 1 corresponds to 0 dB and gain will not be modified. A parameter value of 4 corresponds to 12 dB and an amplification factor of 4.

Setting the Analog Gain Using Basler pylon

To set the Gain Analog All parameter value:

- Set the Gain Selector to Analog All.
- Set the Gain parameter to 1 or 4, as desired.

You can set the Gain Selector and the Gain parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set Gain Analog All
Camera.GainSelector.SetValue( GainSelector_AnalogAll );
Camera.GainRaw.SetValue( 4 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 3.1 on page 15.

Setting the Analog Gain Using Direct Register Access

To set the gain via direct register access:

- Set the value of the Gain Analog All register.

For more information about direct register access, see Section 3.2 on page 16.

8.2.1.2 Digital Gain

Adjusting the camera's digital gain will digitally shift the group of bits that is output for the pixel values from each ADC in the camera.

Increasing the digital gain parameter value will result in an amplified gain and therefore in higher pixel values. Decreasing the digital gain setting will result in a decreased gain and therefore in lower pixel values.

The digital gain parameter values can be set on an integer scale ranging from 256 to 2047. This range of settings is linearly related to a range of amplification factors where a parameter value of 256 corresponds to 0 dB and gain will not be modified and a parameter value of 2047 corresponds to 18.057 dB and an amplification factor of approximately 8.00.

You can use the formula below to calculate the dB of gain that will result from the Gain Raw parameter values:

$$\text{Gain[dB]} = 20 \times \log_{10}\left(\frac{\text{Gain Raw}}{256}\right)$$

$$\text{Gain[dB]} = 20 \times \log_{10}(\text{Gain Raw}) - 48.165$$

Note that due to the nature of digital gain certain gray values will be absent in the image ("missing codes") if digital gain is set to a value larger than 256.

You can use the remove parameter limits feature to remove to lower limit for digital gain parameter values. When you use the remove parameter limits feature you can also set digital gain parameter values in the range from 0 to 256. This corresponds to a range of amplification factors from 0 to approximately 0.99.

> If the digital gain parameter value is set below 256 using the remove parameter limits feature: In this case, regardless of the brightness of illumination, the camera will not be able to reach the maximum gray values that otherwise could be reached. For example, if the camera is set to a 12 bit pixel data format, the maximum gray value of 4095 can not be reached if the digital gain parameter value is set below 256.

For more information about the remove parameter limits feature, see Section 8.3 on page 82.

Setting the Digital Gain Using Basler pylon

To set the Gain All parameter value:

- Set the Gain Selector to All.
- Set the Gain parameter to your desired value.

You can set the Gain Selector and the Gain parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set Gain Digital All
Camera.GainSelector.SetValue( GainSelector_All );
Camera.GainRaw.SetValue( 256 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 3.1 on page 15.

Setting the Analog Gain Using Direct Register Access

To set the gain via direct register access:

- Set the value of the Gain Digital All register.

For more information about direct register access, see Section 3.2 on page 16.

8.2.1.3 Using Both Analog Gain and Digital Gain

You can use analog gain and digital gain at the same time. In this case, the amplification factors will multiply. For example, if you set analog gain to an amplification factor of 4 and use an amplification factor of 1.2 for digital gain the total amplification factor will be 4.8. This corresponds to adding 12 dB and 1.6 dB to give a total gain of 13.6 dB.

For optimum image quality, we recommend to set the total amplification as low as possible. If you need an amplification factor larger than 4 we recommend to set analog gain to 4 and then digital gain to reach the desired total amplification.

ⓘ If you use analog gain and digital gain at the same time and also use the remove parameter limits for digital gain with digital gain parameter values below 256: Note also that the amplification factor for total gain will be 0 if the digital gain setting is 0.

8.2.2 Black Level

Adjusting the camera’s black level will result in an offset to the pixel values output from the camera.

The camera’s black level is determined by the Black Level Raw parameter with the black Level selector set to All. All pixels in the sensor are affected by this setting.

If the camera is set for a pixel data format with an 8 bit depth, an increase of 16 in a black level setting will result in a positive offset of 1 in the pixel values output from the camera. And a decrease of 16 in a black level setting will result in a negative offset of 1 in the pixel values output from the camera.

If the camera is set for a pixel data format with a 12 bit depth, an increase of 1 in a black level setting will result in a positive offset of 1 in the pixel values output from the camera. A decrease of 1 in a black level setting will result in a negative offset of 1 in the pixel values output from the camera.

Setting the Black Level Using Basler pylon

To set the Black Level Raw All value:

- Set the Black Level Selector to All.
- Set the Black Level Raw parameter to your desired value.

You can set the Black Level Selector and the Black Level Raw parameter values from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set Black Level Raw All
Camera.BlackLevelSelector.SetValue ( BlackLevelSelector_All );
Camera.BlackLevelRaw.SetValue( 64 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 3.1 on page 15.

Setting the Black Level Using Direct Register Access

To the black level via direct register access:

- Set the value of the Black Level All register.

For more information about direct register access, see Section 3.2 on page 16.

8.3 Remove Parameter Limits

For each camera feature, the allowed range of any associated parameter values is normally limited. The factory limits are designed to ensure optimum camera operation and, in particular, good image quality. For special camera uses, however, it may be helpful to set parameter values outside of the factory limits.

The remove parameter limits feature lets you remove the factory limits for parameters associated with certain camera features. When the factory limits are removed, the parameter values can be set within extended limits. Typically, the range of the extended limits is dictated by the physical restrictions of the camera's electronic devices, such as the absolute limits of the camera's variable gain control.

The values for any extended limits can be determined by using the Basler pylon Viewer or from within your application via the pylon API.

Currently, the limits can only be removed from the digital gain feature. Removing the parameter limits on the digital gain feature will only remove the lower limit. The lower limit for the DigitalGain parameter is reduced to 0.

For more information about the digital gain feature, see Section 8.2.1.2 on page 79.

Removing Parameter Limits Using Basler pylon

To remove the limits for a parameter:

- Use the Parameter Selector to select the parameter whose limits you want to remove.
- Set the value of the Remove Limits parameter.

You can set the Parameter Selector and the value of the Remove Limits parameter from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Select the feature whose factory limits will be removed.
Camera.ParameterSelector.SetValue( ParameterSelector_Gain );
// Remove the limits for the selected feature.
Camera.RemoveLimits.SetValue( true );
```

You can also use the Basler pylon Viewer application to easily set the parameters. Note that the remove parameter limits feature will only be available at the "guru" viewing level.

Removing Parameter Limits Using Direct Register Access

To disable the limits for the Gain parameter using direct register access:

- Set the value of the Remove Param Limits Gain register to enabled.

For more information about direct register access, see Section 3.2 on page 16.

8.4 Image Area of Interest (AOI)

The image area of interest feature lets you specify a portion of the sensor line. During operation, only the pixel information from the specified portion of the line is read out of the sensor and transmitted from the camera to the frame grabber. One of the main advantages of the AOI feature is that decreasing the size of the AOI can increase the camera's maximum allowed acquisition line rate. For more information about how changing the AOI size affects the maximum allowed line rate, see Section 6.4 on page 66.

The area of interest is referenced to the left end of the sensor array. The outer left pixel is designated as pixel 0 (see Fig. 17).

The location and size of the area of interest is defined by declaring an offset X and a width. For example, suppose that you specify the offset X as 16 and the width as 24. The area of the array that is defined by these settings is shown in Figure 17. With these settings, the camera will read out and transmit pixel values for pixels 16 through 39..

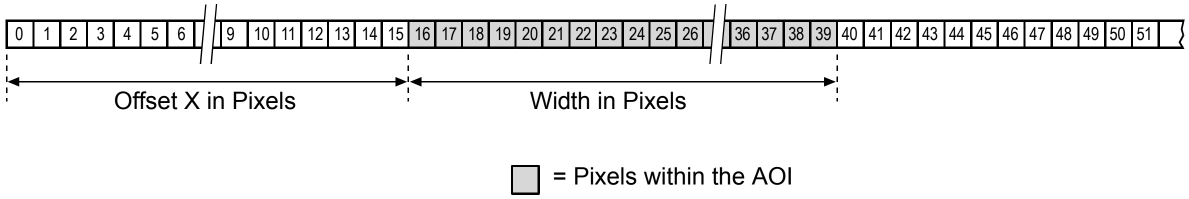


Fig. 17: Image Area of Interest

8.4.1 Setting the Image AOI

By default, the AOI is set to use the full resolution of the camera's sensor. You can change the size and the position of the AOI by changing the value of the camera's Offset X and Height parameters.

When you are setting the camera's area of interest, you must follow these guidelines:

- The sum of the Offset X setting plus the Width setting must not exceed the width of the camera's sensor. For example, on the raL2048-80km, the sum of the Offset X setting plus the Width setting must not exceed 2048.
- Depending on the selected Camera Link tap geometry, the following increments apply to offset X and width if horizontal binning is not enabled:

Camera Link Tap Geometry	Offset X Increment	Width Increment
1X2	8	8
1X4	8	8
1X8	8	8
1X10	8	40

Table 13: Camera Link Tap Geometry and Related Increments for AOI Offset X and Width

For example, if you are working with a camera that has a sensor with 2048 pixels and if you have selected the 1X4 Camera Link tap geometry:

Offset X + AOI Width ≤ 2048

The AOI offset X can be set to 0, 8, 16, 24, etc.

The AOI width can be set to 8, 16, 24, 32, etc.

For more information about Camera Link tap geometries, see Section 7.2 on page 73.

Your frame grabber may place additional restrictions on how the AOI position and size must be set. Check the documentation included with your frame grabber to determine its AOI requirements.

Normally, the X Offset and Width parameter settings refer to the physical line of the sensor. But if binning is enabled, these parameters are set in terms of a "virtual" line. For more information about using an image AOI with binning, see Section 8.6 on page 89.

Setting the Image AOI Using Basler pylon

You can set the Offset X and Width parameter values from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to get the maximum allowed settings and the increments for the Width and Height parameters. They also illustrate setting the Offset X and Width parameter values.

```
int64_t widthMax = Camera.Width.GetMax( );
int64_t widthInc = Camera.Width.GetInc();
int64_t offsetxMax = Camera.OffsetX.GetMax( );
int64_t offsetxInc = Camera.OffsetX.GetInc();
Camera.Width.SetValue( 24 );
Camera.OffsetX.SetValue( 16 );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting the Image AOI Using Direct Register Access

To set the AOI Offset X and Width parameter values via direct register access:

- Set the value of the Offset X register.
- Set the value of the Width register.

Using the Image AOI Feature with Horizontal Binning Enabled

If the camera’s horizontal binning feature (see Section 8.6 on page 89) is enabled, it will have an effect on the way that you set up the area of interest. With horizontal binning enabled, pairs of adjacent pixel in a line are binned together into a single pixel. When you set an AOI with horizontal binning enabled, the AOI settings are based on "binned" pixels rather than the actual physical pixels in the sensor lines.

- If you are working with a camera that has sensor lines with 2048 physical pixels and you enable horizontal binning by 2, you effectively reduce the resolution to 1024 "binned" pixels. In this case, you would set the AOI as if you were working with a sensor that actually has 1024 pixels.
- If you are working with a camera that has sensor lines with 2048 physical pixels and you enable horizontal binning by 4, you effectively reduce the resolution to 512 "binned" pixels. In this case, you would set the AOI as if you were working with a sensor that actually has 512 pixels.
- If you are working with a camera that has sensor lines with 4096 physical pixels and you enable horizontal binning by 2, you effectively reduce the resolution to 2048 "binned" pixels. In this case, you would set the AOI as if you were working with a sensor that actually has 2048 pixels.
- If you are working with a camera that has sensor lines with 4096 physical pixels and you enable horizontal binning by 4, you effectively reduce the resolution to 1024 "binned" pixels. In this case, you would set the AOI as if you were working with a sensor that actually has 1024 pixels.
- If you are using binning by 3, make sure to set up an area of interest where the With parameter value is evenly divisible by 3.

8.5 Luminance Lookup Table

The type of electronics used on the camera allow the camera's sensor to acquire pixel values at a 12 bit depth. Normally, when a camera is set for a 12 bit pixel data format, the camera uses the actual 12 bit pixel values reported by the sensor.

The luminance lookup table feature lets you create a custom 12 bit to 12 bit lookup table that maps the actual 12 bit values output from the sensor to substitute 12 bit values of your choice. When the lookup table is enabled, the camera will replace the actual pixel values output from the sensor with the substitute values from the table.

The lookup table has 4096 indexed locations with a 12 bit value stored at each index. The values stored in the table are used like this:

- When the sensor reports that a pixel has an actual 12 bit value of 0, the substitute 12 bit value stored at index 0 will replace the actual pixel value.
- The numbers stored at indices 1 through 7 are not used.
- When the sensor reports that a pixel has an actual 12 bit value of 8, the substitute 12 bit value stored at index 8 will replace the actual pixel value.
- The numbers stored at indices 9 through 15 are not used.
- When the sensor reports that a pixel has an actual 12 bit value of 16, the substitute 12 bit value stored at index 16 will replace the actual pixel value.
- The numbers stored at indices 17 through 23 are not used.
- When the sensor reports that a pixel has an actual 12 bit value of 24, the substitute 12 bit value stored at index 24 will replace the actual pixel value.
- And so on.

As you can see, the table does not include a defined 12 bit substitute value for every actual pixel value that the sensor can report. If the sensor reports an actual pixel value that is between two values that have a defined substitute, the camera performs a straight line interpolation to determine the substitute value that it should use. For example, assume that the sensor reports an actual pixel value of 12. In this case, the camera would perform a straight line interpolation between the substitute values at index 8 and index 16 in the table. The result of the interpolation would be used by the camera as the substitute.

Another thing to keep in mind about the table is that index 4088 is the last index that will have a defined substitute value associated with it (the values at indices 4089 through 4095 are not used.) If the sensor reports an actual value greater than 4088, the camera will not be able to perform an interpolation. In cases where the sensor reports an actual value greater than 4088, the camera simply uses the 12 bit substitute value from index 4088 in the table.

The advantage of the luminance lookup table feature is that it lets a user customize the response curve of the camera. The graphs below represent the contents of two typical lookup tables. The first graph is for a lookup table where the values are arranged so that the output of the camera increases linearly as the actual sensor output increases. The second graph is for a lookup table where the values are arranged so that the camera output increases quickly as the actual sensor output moves from 0 through 2048 and increases gradually as the actual sensor output moves from 2049 through 4096.

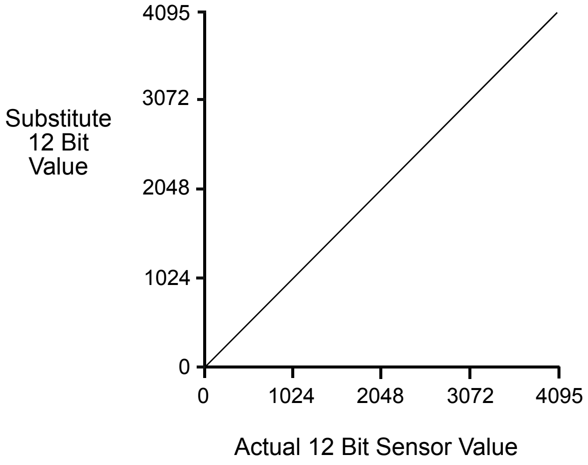


Fig. 18: Lookup Table with Values Mapped in a Linear Fashion

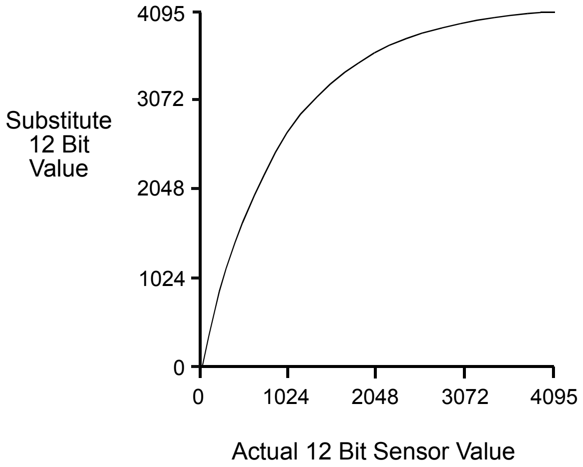


Fig. 19: Lookup Table with Values Mapped for Higher Camera Output at Low Sensor Readings

Using the Luminance Lookup Table to Get 8 Bit Output

As mentioned above, when the camera is set for a 12 bit pixel data format, the lookup table can be used to perform a 12 bit to 12 bit substitution. The lookup table can also be used in 12 bit to 8 bit fashion. To use the table in 12 bit to 8 bit fashion, you enter 12 bit substitution values into the table and enable the table as you normally would. But instead of setting the camera for a 12 bit pixel data format, you set the camera for an 8 bit format (such as Mono 8). In this situation, the camera will first use the values in the table to do a 12 bit to 12 bit substitution. It will then truncate the lowest 4 bits of the substitute value and will transmit the remaining 8 highest bits.

Changing the Values in the Luminance Lookup Table and Enabling the Table Using Basler pylon

You can change the values in the luminance lookup table (LUT) and enable the use of the lookup table by doing the following:

1. Use the LUT Selector to select a lookup table. (Currently there is only one lookup table available, i.e., the "luminance" lookup table described above.)
2. Use the LUT Index parameter to select an index number.
3. Use the LUT Value parameter to enter the substitute value that will be stored at the index number that you selected in step 2.
4. Repeat steps 2 and 3 to enter other substitute values into the table as desired.
5. Use the LUT Enable parameter to enable the table.

You can set the LUT Selector, the LUT Index parameter and the LUT Value parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter values:

```
// Select the lookup table
Camera.LUTSelector.SetValue( LUTSelector_Luminance );

// Write a lookup table to the device.
// The following lookup table causes an inversion of the sensor values
// ( bright -> dark, dark -> bright )
for ( int i = 0; i < 4096; i += 8 )
{
   Camera.LUTIndex.SetValue( i );
   Camera.LUTValue.SetValue( 4095 - i );
}
// Enable the lookup table
Camera.LUTEnable.SetValue( true );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

Changing the Values in the Luminance Lookup Table and Enabling the Table Using Direct Register Access

When setting up the luminance lookup table via direct register access, two registers are involved: the LUT register and the LUT Enable register.

- The LUT register is simply an array register that holds the 4096 12 bit values described earlier in this section. As a first step to using the lookup table feature. You must populate this register with 12 bit values.
- Once the LUT register has been populated, you can enable the use of the lookup table by setting the value of the LUT Enable register to 1 (enabled).

For more information about direct register access, see Section 3.2 on page 16.

8.6 Binning

Binning increases the camera's response to light by summing the charges from adjacent pixels into one pixel.

With horizontal binning, the charges of 2, 3, or a maximum of 4 adjacent pixels are summed and are reported out of the camera as a single pixel. Fig. 20 illustrates horizontal binning.

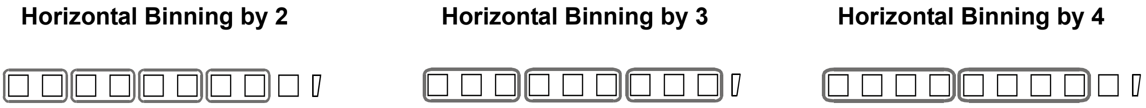


Fig. 20: Horizontal Binning

For information about using binning with the Image Area of Interest feature enabled, see Section 8.4 on page 83.

Setting Binning Using Basler pylon

You can enable horizontal binning by setting the Binning Horizontal parameter. Setting the parameter's value to 2, 3, or 4 enables horizontal binning by 2, horizontal binning by 3, or horizontal binning by 4 respectively. Setting the parameter's value to 1 disables horizontal binning.

You can set the Binning Vertical or the Binning Horizontal parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values:

```
// Enable horizontal binning by 4
Camera.BinningHorizontal.SetValue( 4 );

// Disable horizontal binning
Camera.BinningHorizontal.SetValue( 1 );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting Binning Using Direct Register Access

To enable horizontal binning via direct register access:

- Set the value of the Binning Horizontal register.

Setting the register's value to 2, 3, or 4 enables horizontal binning by 2, horizontal binning by 3, or horizontal binning by 4 respectively. Setting the register's value to 1 disables horizontal binning.

For more information about direct register access, see Section 3.2 on page 16.

8.7 Gamma Correction

The gamma correction feature lets you modify the brightness of the pixel values output by the camera's sensor to account for a non-linearity in the human perception of brightness. To accomplish the correction, a gamma correction factor (γ) is applied to the brightness value (Y) of each pixel according to the following formula:

$$Y_{corrected} = \left(\frac{Y_{uncorrected}}{Y_{max}}\right)^{\gamma} \times Y_{max}$$

The formula uses uncorrected and corrected pixel brightnesses that are normalized by the maximum pixel brightness. The maximum pixel brightness equals 255 for 8 bit output and 4095 for 12 bit output.

When the gamma correction factor is set to 1, the output pixel brightness will not be corrected.

A gamma correction factor between 0 and 1 will result in increased overall brightness, and a gamma correction factor greater than 1 will result in decreased overall brightness.

In all cases, black (output pixel brightness equals 0) and white (output pixel brightness equals 255 at 8 bit output and 4095 at 12 bit output) will not be corrected.

Enabling Gamma Correction and Setting the Gamma Using Basler pylon

You can enable or disable the gamma correction feature by setting the value of the Gamma Enable parameter.

When gamma correction is enabled, the correction factor is determined by the value of the Gamma parameter. The Gamma parameter can be set in a range from 0 to 3.99902. So if the Gamma parameter is set to 1.2, for example, the gamma correction factor will be 1.2.

You can set the Gamma Enable and Gamma parameter values from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values:

```
// Enable the Gamma feature
Camera.GammaEnable.SetValue( true );

// Set the Gamma value to 1.2
Camera.Gamma.SetValue( 1.2 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Enabling Gamma Correction and Setting the Gamma Using Direct Register Access

To enable gamma correction and to set the gamma value via direct register access:

- Set the value of the Gamma Enable register to Enabled.
- Set the value of the Gamma Selector register to User.
- If the Gamma Selector is set to User, set the value in the Gamma register to the desired gamma value.

For more information about direct register access, see Section 3.2 on page 16.

8.8 Shading Correction

Two types of shading correction are available on the camera, offset shading correction and gain shading correction. You can set the camera to only perform offset shading correction, to only perform gain shading correction, or to perform both types of shading correction.

8.8.1 Offset Shading Correction

When a line scan camera with a digital sensor captures a line in complete darkness, all of the pixel values in the line should be near zero and they should be equal. In practice, slight variations in the performance of the pixels in the sensor will cause some variation in the pixel values output from the camera when the camera is capturing lines in darkness. Offset shading correction (also known as dark signal non-uniformity (DSNU) correction) corrects for this type of variation.

Offset shading correction works by adding an individual gray value to each pixel value in the acquired lines. The gray values used for correction are included in a "shading file", commonly also referred to as a "shading set". In order to use offset shading correction, the user must enable offset shading correction and activate the related defaultshading file or the previously created usershading file (see below).

8.8.2 Gain Shading Correction

When a line scan camera with a digital sensor captures a line with the camera viewing a uniform light-colored target in bright light, all of the pixel values in the line should be near their maximum gray value and they should be equal. In practice, slight variations in the performance of the pixels in the sensor, variations in the optics, and variations in the lighting will cause some variation in the pixel values output from the camera. Gain shading correction (also known as photo response non-uniformity (PRNU) correction) corrects for this type of variation.

Gain shading correction works by applying an individual multiplier to each pixel value in the acquired lines. The multipliers used for correction are included in a "shading file", commonly also referred to as a "shading set". In order to use gain shading correction, the user must enable gain shading correction and activate the related defaultshading file or the previously created usershading file (see below).

8.8.3 Default Shading Set File and User Shading Set File

For each type of shading correction, two types of shading set files are available in the camera’s nonvolatile memory:

- The first type of shading set file is called the "defaultshading" file. One "defaultshading" file is available for offset shading correction and another one for gain shading correction.

  A "defaultshading" file contains a complete collection of the values needed to perform either offset shading or gain shading correction. The values in the files are generated during the camera’s factory setup procedure and they essentially serve as default shading correction values. The values are optimized for performing shading correction with "standard" optics and lighting. Using the factory values will give you reasonable shading correction performance in most situations. One advantage of the factory values is that they serve as a good default. The "defaultshading" files are in a protected area of the camera’s memory and can’t be changed.
- The second type of shading set file is called the "usershading" file. One "usershading" file is available for offset shading correction and another one for gain shading correction.

  Like a "defaultshading" file, a "usershading" file also holds a complete collection of the values needed to perform either offset shading or gain shading correction. The values stored in the files must, however, be generated by the camera user. When the values are generated the camera must operate under its real world conditions. The "usershading" files contain the shading correction values that will normally be used for day-to-day camera operation. A procedure describing how to generate the values in the files appears below.

8.8.3.1 Creating a "Usershading" File

To create a "usershading" file and enable it, you must take the steps listed below. We strongly recommend that you read through all of the steps and read all of the other information in this section before you attempt to do shading correction.

The steps below are intended to give you the basic knowledge needed to create a "usershading" file and to activate it. **A code sample that includes the complete details of how to create a usershading file and how to enable shading correction on a camera is included with the Basler pylon SDK.**

The steps also include information about using direct register access for creating and activating a "usershading" file.

When you create a "usershading" file you must make sure to create correction values for all of the pixels in the sensor’s line regardless of how you plan to use the camera during normal operation.

Creating a "Usershading" File for Offset Shading Correction

ⓘ	Creating a "usershading" file for offset shading correction will overwrite any "usershading" file for offset shading correction that is already in the camera’s memory. If you want to preserve the previous "usershading" file save it to your PC before creating the new "usershading" file. For information about saving a "usershading" file to the PC, see Section 8.8.3.2 on page 96.

To create a "usershading" file for offset shading correction, perform the following steps:

1. Adjust the lighting, optics, line rate, exposure time control mode, exposure time, gain, and camera temperature as you would for normal operation.
2. Prevent light from striking the camera’s sensor: Cover the camera lens, close the iris in the camera lens, or darken the room so that the camera will be capturing lines in complete darkness.
3. Set the camera’s X Offset and Width parameters so that the entire width of the sensor will be used during line acquisition.
   Note: You can create a "usershading" file for an AOI that is narrower than the entire width of the sensor. In this case, however, the "usershading" file will only apply to the narrower AOI or to smaller included AOIs. We recommend using the entire width of the sensor.
4. If you use Basler pylon, select Offset Shading in the Basler pylon Viewer or via the Basler pylon API.
5. If you use Basler pylon, select "usershading" file in the Basler pylon Viewer or via the Basler pylon API.

   If you use direct register access, set the value of the Offset Shading Set Selector register to the desired user set, e.g. User Shading Set 1.
6. If you use Basler pylon, go to the **Create** enumeration in the Basler pylon Viewer and select **Once** or send a create command via the Basler pylon API.

   If you use direct register access, set the value of the Offset Shading Create register to OnePush.
7. Perform at least 128 line acquisitions.

   For more information about the line start trigger mode, see Section 6.1 on page 43.

   After 128 line acquisitions are completed the camera creates the "usershading" file automatically. The "usershading" file is stored in the camera’s non-volatile memory and is not lost if the camera power is switched off.

ⓘ	Any time you make a change to the line rate, exposure time control mode, exposure time, gain, or camera temperature, you must create a new "usershading" file for offset shading correction. Using an out of date "usershading" file can result in poor image quality.

Creating a "Usershading" File for Gain Shading Correction

Creating a "usershading" file for gain shading correction will overwrite any "usershading" file for gain shading correction that is already in the camera’s memory.

If you want to preserve the previous "usershading" file save it to your PC before creating the new "usershading" file.

For information about saving a "usershading" file to the PC, see Section 8.8.3.2 on page 96.

To create a "usershading" file for gain shading correction, perform the following steps:

1. Adjust the lighting, optics, line rate, exposure time control mode, exposure time, gain, and camera temperature as you would for normal operation.
2. Place a uniform white target in the field of view of the camera.
3. Set the camera’s X Offset and Width parameters so that the entire width of the sensor lines will be used during line acquisition.
4. Perform several line acquisitions and examine the pixel values returned from the camera. The pixel values for the brightest pixels in each line should be about 90 to 95 % of maximum (i.e., if the camera is set for 8 bit output, the pixels should be from 90 to 95 % of 255).
   a. If the values for the brightest pixels are at 90 to 95 % of maximum, go on to step 5.
   b. If the values for the brightest pixels are not at 90 to 95 % of the maximum, adjust your lighting and/or lens aperture to achieve 90 to 95 %
5. Perform several line acquisitions and examine the pixel values returned from the camera. In each line, the values for the darkest pixels must be greater than 1/4 of the values for the brightest pixels. (If the values for the darkest pixels are less than 1/4 of the values for the brightest, the camera will not be able to fully correct for shading variations.)
   a. If the values for the darkest pixels are greater than 1/4 of the values for the brightest, go on to step 6.
   b. If the values for the darkest pixels are less than 1/4 of the values for the brightest pixels, it usually indicates extreme variations in lighting or poor quality optics. Make corrections as required.
6. If you use Basler pylon, select Gain Shading in the Basler pylon Viewer or via the Basler pylon API.
7. If you use Basler pylon, select "usershading" file in the Basler pylon Viewer or via the Basler pylon API.

   If you use direct register access, set the value of the Gain Shading Set Selector register to the desired user set, e.g. User Shading Set 1
8. If you use Basler pylon, go to the **Create** enumeration in the Basler pylon Viewer and select **Once** or send a create command via the Basler pylon API.

   If you use direct register access, set the value of the Gain Shading Create register to OnePush
9. Perform at least 128 line acquisitions.

For more information about the line start trigger mode, see Section 6.1 on page 43.

After 128 line acquisitions are completed the camera creates the "usershading" file automatically. The "usershading" file is stored in the camera’s non-volatile memory and is not lost if the camera power is switched off.

Any time you make a change to the optics or lighting or if you change the camera’s gain settings or exposure mode, you must create a new "usershading" file. Using an out of date "usershading" file can result in poor image quality.

8.8.3.2 Working with Shading Sets Using Basler pylon

Once you have created shading set files, you can use the following pylon API functions to work with the shading sets:

Shading Selector - is used to select the type of shading correction to configure, i.e. offset shading correction or gain shading correction.

Shading Create - is used to create a "usershading" file. The enumeration allows selecting the settings **Off** and **Once**.

Shading Enable - is used to enable and disable the selected type of shading correction.

Shading Set Selector - is used to select the shading set to which the activate and the create enumeration commands will be applied.

Shading Set Activate - is used to activate the selected shading set. "Activate" means that the shading set will be copied from the camera’s non-volatile memory into its volatile memory. When the shading correction feature is enabled, the shading set in the volatile memory will be used to perform shading correction.

Shading Set Default Selector - is used to select the shading set that will be loaded into the camera’s volatile memory during camera bootup.

Shading Status - is used to determine the error status of operations such as Shading Set Activate. The following error statuses may be indicated:

No error - the last operation performed was successful.

Startup Set error - there was a problem with the default shading set.

Activate error - the selected shading set could not be loaded into the volatile memory.

Create error - an error occurred during the attempt of creating a "usershading" file.

The use of the pylon API functions listed above is illustrated in the shading correction sample code included with the pylon SDK.

You can also use the Shading parameters group in the Basler pylon Viewer application to access these functions.

And you can use the File Access selection in the Camera menu of the Viewer to save a shading set file to a PC and to upload a shading set file from the PC to the camera.

8.8.3.3 Working with Shading Sets Using Direct Register Access

Once you have created shading set files, you can use the following registers to work with the shading sets:

Offset Shading Enable/Gain Shading Enable - is used to enable and disable offset or gain shading correction.

Offset Shading Set Selector/Gain Shading Set Selector - is used to select the shading set to which the activate and the create commands will be applied.

Offset Shading Create/Gain Shading Create - is used to create a "usershading" file. The enumeration allows selecting the settings Off and OnePush.

Offset Shading Activate/Gain Shading Activate - is used to activate the selected shading set. "Activate" means that the shading set will be copied from the camera's non-volatile memory into its volatile memory. When the shading correction feature is enabled, the shading set in the volatile memory will be used to perform shading correction.

Offset Shading Startup Set/Gain Shading Startup Set - is used to select the shading set that will be loaded into the camera's volatile memory during camera bootup.

Offset Shading Status/Gain Shading Status - is used to determine the error status of operations such as Offset Shading Activate. The following error statuses may be indicated:

No error - the last operation performed was successful.

Startup Set error - there was a problem with the default shading set.

Activate error - the selected shading set could not be loaded into the volatile memory.

Create error - an error occurred during the attempt of creating a "usershading" file.

For more information about direct register access, see Section 3.2 on page 16.

8.9 Error Detection

8.9.1 LED Indicator

The LED indicator on the back of the camera includes both a small red LED and a small green LED. The LED indicator signals the camera's current condition as shown in Table 14.

LED State	Status Indication
Red and Green Both Off	No power to the camera
Continuous Green	The camera has booted up successfully and is OK.
Flashing Green	The camera is set to expect an external trigger signal on an input, but no trigger signal is present.
Green and Red Both Flashing Alternately	An error condition has been detected that may be correctable with user intervention. (See Section 8.9.2 on page 99 for more information).
Flashing Red	Internal error. Contact Basler technical support.

Table 14: LED Indications

During the camera bootup process, the red and the green LED will simultaneously be lit.

8.9.2 Error Codes

The camera can detect several user correctable errors. If one of these errors is present, the camera will set an error code and will flash alternately both the red and green LEDs in the LED indicator.

The following table indicates the available error codes:

Code	Condition	Meaning
0	No Error	The camera has not detected any errors since the last time that the error memory was cleared.
1	Overtrigger	An overtrigger has occurred. The user has applied a line start trigger to the camera when the camera was not in a waiting for line start trigger condition.
2	User Set Load	An error occurred when attempting to load a user set. Typically, this means that the user set contains an invalid value. Try loading a different user set.
3	Invalid Parameter	A parameter is set out of range or in an otherwise invalid manner. (Typically, this error only occurs when the user is setting parameters via direct register access.)

Table 15: Error Codes

When the camera detects a user correctable error, it sets the appropriate error code in an error memory. If two or three different detectable errors have occurred, the camera will store the code for each type of error that it has detected (it will store one occurrence of the each code no matter how many times it has detected the corresponding error).

You can use the following procedure to check the error codes:

- Read the value of the Last Error parameter. The Last Error parameter will indicate the last error code stored in the memory.
- Execute the Clear Last Error Command to clear the last error code from the memory.
- Continue reading and clearing the last error until the parameter indicates a No Error code.

Reading and Clearing the Error Codes Using Basler pylon

You can use the pylon API to read the value of the Last Error parameter and to execute a Clear Last Error command from within your application software. The following code snippets illustrate using the API to read the parameter value and execute the command:

```
// Read the value of the last error code in the memory
LastErrorEnums lasterror = Camera.LastError.GetValue();

// Clear the value of the last error code in the memory
Camera.ClearLastError.Execute( );
```

You can also use the Basler pylon Viewer application to easily set the parameter and execute the command.

For more information about the pylon API and the pylon Viewer, see Section 3.1 on page 15.

Reading and Clearing the Error Codes Using Direct Register Access

To get the value of the last error code in the memory via direct register access:

- Read the value of the Last User Error register.

To clear the value of the last error code in the memory via direct register access:

- Set the value of the Clear Last User Error register to 1.

For more information about direct register access, see Section 3.2 on page 16.

8.10 Test Images

All cameras include the ability to generate test images. Test images are used to check the camera's basic functionality and its ability to transmit an image to the host PC. Test images can be used for service purposes and for failure diagnostics.

When the camera is in test image mode, the optics and the imaging sensor are not used. The lines that make up each test image are generated internally by the camera's logic and are transmitted to the host PC line by line.

> ⓘ If the camera is set to use an electrical signal applied to input line CC1, CC2, CC3 or CC4 as the source signal for the line trigger, these signals must be provided to the camera in order to generate test images.

The Effect of Camera Settings on Test Images

When any test image is active, the camera's analog features such as analog gain, black level, and exposure time have no effect on the images transmitted by the camera. For test images 1, 2, and 3, the camera's digital features, will also have no effect on the transmitted images. But for test images 4 and 5, the cameras digital features will affect the images transmitted by the camera.

Make sure to disable shading correction before generating test images.

Enabling a Test Image Using Basler pylon

The Test Image Selector is used to set the camera to output a test image. You can set the value of the Test Image Selector to one of the test images or to "test image off".

You can set the Test Image Selector from within your application software by using the pylon API. The following code snippets illustrate using the API to set the selector:

```
// set for no test image
Camera.TestImageSelector.SetValue( TestImageSelector_Off );

// set for the first test image
Camera.TestImageSelector.SetValue( TestImageSelector_Testimage1 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 3.1 on page 15.

Enabling a Test Image Using Direct Register Access

To enable a test image via direct register access:

- Set the value of the Test Image Selector register to Test Image 1, 2, 3, 4, or 5 as desired.

To disable test images:

- Set the value of the Test image Selector Register to Off.

For more information about direct register access, see Section 3.2 on page 16.

8.10.1 Test Images in Detail

Test Image 1 - Fixed Horizontal Gray Gradient (8 bit)

The 8 bit fixed horizontal gray gradient test image is best suited for use when the camera is set for monochrome 8 bit output. The test image consists of fixed horizontal gray gradients ranging from 0 to 255.

The mathematical expression for this test image is:

Gray Value = [column number] MOD 256

Test Image 2 - Moving Diagonal Gray Gradient (8 bit)

The 8 bit moving diagonal gray gradient test image is similar to test image 1, but it is not stationary. The image moves by one pixel from right to left whenever a new line acquisition is initiated. The test pattern uses a counter that increments by one for each new line acquisition.

The mathematical expression for this test image is:

Gray Value = [column number + counter] MOD 256

Test Image 3 - Moving Diagonal Gray Gradient (12 bit)

The 12 bit moving diagonal gray gradient test image is similar to test image 2, but it is a 12 bit pattern. The image moves by one pixel from right to left whenever a new line acquisition is initiated. The test pattern uses a counter that increments by one for each new line acquisition.

The mathematical expression for this test image is:

Gray Value = [column number + counter] MOD 4096

Test Image 4 - Moving Diagonal Gray Gradient Feature Test (8 bit)

The basic appearance of test image 4 is similar to test image 2 (the 8 bit moving diagonal gray gradient image). The difference between test image 4 and test image 2 is this: if a camera feature that involves digital processing is enabled, test image 4 **will** show the effects of the feature while test image 2 **will not**. This makes test image 4 useful for checking the effects of digital features such as digital gain.

Test Image 5 - Moving Diagonal Gray Gradient Feature Test (12 bit)

The basic appearance of test image 5 is similar to test image 3 (the 12 bit moving diagonal gray gradient image). The difference between test image 5 and test image 3 is this: if a camera feature that involves digital processing is enabled, test image 5 **will** show the effects of the feature while test image 3 **will not**. This makes test image 5 useful for checking the effects of digital features.

8.11 Device Information Parameters

Each camera includes a set of "device information" parameters. These parameters provide some basic information about the camera. The device information parameters include:

- Device Vendor Name (read only) - contains the name of the camera’s vendor. This string will always indicate Basler as the vendor.
- Device Model Name (read only) - contains the model name of the camera, for example, raL2048-80km.
- Device Manufacturer Info (read only) - can contain some information about the camera manufacturer. This string usually indicates "none".
- Device Version (read only) - contains the device version number for the camera.
- Firmware Version (read only) - contains the version of the firmware in the camera.
- Device ID (read only) - contains the serial number of the camera.
- Device User ID (read / write) - is used to assign a user defined name to a device. This name will be displayed in the Basler pylon Viewer and the Basler pylon IP Configuration Tool. The name will also be visible in the "friendly name" field of the device information objects returned by pylon’s device enumeration procedure.
- Device Scan Type (read only) - contains the scan type of the camera (line scan).
- Sensor Width (read only) - contains the physical width of the sensor in pixels.
- Sensor Height (read only) - contains the physical height of the sensor in pixels.
- Max Width (read only) - Indicates the camera’s maximum width setting.
- Max Height (read only) - Indicates the camera’s maximum height setting.
- Sensor Board Temperature (read only) - Indicates the current temperature in degrees centigrade of the camera’s sensor board.

Working with Device Information Parameters Using Basler pylon

You can read the values for all of the device information parameters or set the value of the Device User ID parameter from within your application software by using the pylon API. The following code snippets illustrate using the API to read the parameters or write the Device User ID:

```
// Read the Vendor Name parameter
Pylon::String_t vendorName = Camera.DeviceVendorName.GetValue();

// Read the Model Name parameter
Pylon::String_t modelName = Camera.DeviceModelName.GetValue();

// Read the Manufacturer Info parameter
Pylon::String_t manufacturerInfo = Camera.DeviceManufacturerInfo.GetValue();

// Read the Device Version parameter
Pylon::String_t deviceVersion = Camera.DeviceVersion.GetValue();
```

```
// Read the Firmware Version parameter
Pylon::String_t firmwareVersion = Camera.DeviceFirmwareVersion.GetValue();

// Read the Device ID parameter
Pylon::String_t deviceID = Camera.DeviceID.GetValue();

// Write and read the Device User ID
Camera.DeviceUserID = "custom name";
Pylon::String_t deviceUserID = Camera.DeviceUserID.GetValue();

// Read the Sensor Width parameter
int64_t sensorWidth = Camera.SensorWidth.GetValue();

// Read the Sensor Height parameter
int64_t sensorHeight = Camera.SensorHeight.GetValue();

// Read the Max Width parameter
int64_t maxWidth = Camera.WidthMax.GetValue();

// Read the Max Height parameter
int64_t maxHeight = Camera.HeightMax.GetValue();

// Read the Temperature Abs parameter
camera.TemperatureSelector.SetValue(TemperatureSelector_Sensorboard);
double temperature = camera.TemperatureAbs.GetValue();
```

You can also use the Basler pylon Viewer application to easily read the parameters and to read or write the Device User ID.

You can use the Basler pylon IP Configuration tool to read or write the Device User ID.

For more information about the pylon Viewer, see Section 3.1 on page 15.

For detailed information about using the pylon API and the pylon IP Configuration Tool, refer to the Basler pylon Programmer’s Guide and API Reference.

Working with Device Information Parameters Using Direct Register Access

When working with the camera via direct register access, you can do the following:

Register Name	Read the Value	Write the Value
Device Vendor Name	●	
Device Model Name	●	
Device Manufacturer Info	●	
Device Version	●	
Device Firmware Version	●	
Device ID	●	
Device User ID	●	●
Device Scan Type	●	
Sensor Width	●	
Sensor Height	●	
Width Max	●	
Height Max	●	

Table 16: Working with Registers Related to Device Information Parameters

For more information about direct register access, see Section 3.2 on page 16.

8.12 User Defined Values

The camera can store two "user defined values". These two values are 32 bit signed integer values that you can set and read as desired. They simply serve as convenient storage locations for the camera user and have no impact on the operation of the camera.

The two values are designated as Value 1 and Value 2.

Setting User Defined Values Using Basler pylon

Setting a user defined value using Basler pylon is a two step process:

- Set the User Defined Value Selector to Value 1 or Value 2.
- Set the User Defined Value parameter to the desired value for the selected value.

You can use the pylon API to set the User Defined Value Selector and the User Defined Value parameter value from within your application software. The following code snippet illustrates using the API to set the selector and the parameter value:

```
// Set user defined value 1
Camera.UserDefinedValueSelector.SetValue( UserDefinedValueSelector_Value1 );
Camera.UserDefinedValue.SetValue( 1000 );

// Set user defined value 2
Camera.UserDefinedValueSelector.SetValue( UserDefinedValueSelector_Value2 );
Camera.UserDefinedValue.SetValue( 2000 );

// Get the value of user defined value 1
Camera.UserDefinedValueSelector.SetValue( UserDefinedValueSelector_Value1 );
int64_t UserValue1 = Camera.UserDefinedValue.GetValue();
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the Basler pylon API and the pylon Viewer, see Section 3.1 on page 15.

Setting User Defined Values Using Direct Register Access

To set the user defined values via direct register access:

- Set the value of the User Defined Value 1 register or the User Defined Value 2 register as desired.

For more information about direct register access, see Section 3.2 on page 16.

8.13 Configuration Sets

A configuration set is a group of values that contains all of the parameter settings needed to control the camera. There are three basic types of configuration sets: the active configuration set, the default configuration set, and user configuration sets.

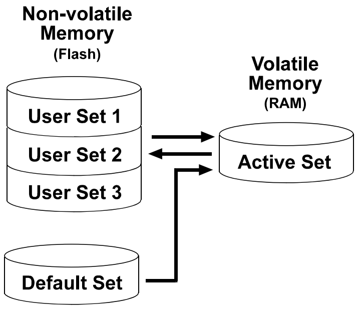


Fig. 21: Configuration Sets

Active Configuration Set

The active configuration set contains the camera's current parameter settings and thus determines the camera's performance, that is, what your image currently looks like. When you change parameter settings using the pylon API or the pylon Viewer, you are making changes to the active configuration set. The active configuration set is located in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off. The active configuration set is usually called the "active set" for short.

Default Configuration Set

When a camera is manufactured, a test setup is performed on the camera and an optimized configuration is determined. The default configuration set contains the camera's factory optimized configuration. The default configuration set is saved in a permanent file in the camera's non-volatile memory. It is not lost when the camera is reset or switched off and it cannot be changed. The default configuration set is usually just called the "default set" for short.

User Configuration Sets

As mentioned above, the active configuration set is stored in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off. The camera can save most of the settings from the current active set to a reserved area in the camera's non-volatile memory. A configuration set saved in the non-volatile memory is not lost when the camera is reset or switched off. There are three reserved areas in the camera's non-volatile memory available for saving configuration sets. A configuration set saved in a reserved area is commonly referred to as a "user configuration set" or "user set" for short.

The three available user sets are called User Set 1, User Set 2, and User Set 3.

ⓘ	The settings for the luminance lookup table are not saved in the user sets and are lost when the camera is reset or switched off. If used, these settings must be set again after each camera reset or restart.

Default Startup Set

You can select the default configuration set or one of the user configuration sets stored in the camera's non-volatile memory to be the "default startup set." The configuration set that you designate as the default startup set will be loaded into the active set whenever the camera starts up at power on or after a reset. Instructions for selecting the default startup set appear in Section 8.13.3 on page 111.

8.13.1 Saving Configuration Sets

Saving the current active set into a user set in the camera's non-volatile memory is a three step process:

- Make changes to the camera's settings until the camera is operating in a manner that you would like to save.
- Set the User Set Selector to User Set 1, User Set 2, or User Set 3.
- Execute a User Set Save command to save the active set to the selected user set.

Saving an active set to a user set in the camera's non-volatile memory will overwrite any parameters that were previously saved in that user set.

Saving User Sets Using Basler pylon

You can set the User Set Selector and execute the User Set Save command from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue( UserSetSelector_UserSet1 );
Camera.UserSetSave.Execute( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 3.1 on page 15.

Saving User Sets Using Direct Register Access

To save the current active set to a user set in the camera's non-volatile memory via direct register access:

- Make changes to the camera's settings until the camera is operating in a manner that you would like to save.
- Set the value of the User Set Selector register to User Set 1, 2, or 3 as desired.
- Set the value of the User Set Save register to 1.

For more information about direct register access, see Section 3.2 on page 16.

8.13.2 Loading a Saved Set or the Default Set into the Active Set

If you have saved a configuration set into the camera's non-volatile memory, you can load the saved set from the camera's non-volatile memory into the camera's active set. When you do this, the loaded set overwrites the parameters in the active set. Since the settings in the active set control the current operation of the camera, the settings from the loaded set will now be controlling the camera.

Loading a Set Using Basler pylon

You can also load the default set into the camera's active set.

To load a saved configuration set or the default set from the camera's non-volatile memory into the active set:

- Set the User Set Selector to User Set 1, User Set 2, User Set 3, or Default.
- Execute a User Set Load command to load the selected set into the active set.

You can set the User Set Selector and execute the User Set Load command from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue( UserSetSelector_UserSet2 );
Camera.UserSetLoad.Execute( );
```

ⓘ Loading a user set or the default set into the active set is only allowed when the camera is idle, i.e. when it is not acquiring lines.

Loading the default set into the active set is a good course of action if you have grossly misadjusted the settings in the camera and you are not sure how to recover. The default settings are optimized for use in typical situations and will provide good camera performance in most cases.

Loading a Set Using Direct Register Access

To load a saved user set or the default set from the camera's non-volatile memory into the active set via direct register access:

- Set the value of the User Set Selector register to User Set 1, 2, or 3, or to the Default set as desired.
- Set the value of the User Set Load register to 1.

For more information about direct register access, see Section 3.2 on page 16.

8.13.3 Selecting the Default Startup Set

You can select the default configuration set or one of the user configuration sets stored in the camera's non-volatile memory to be the "default startup set". The configuration set that you designate as the default startup set will be loaded into the active set whenever the camera starts up at power on or after a reset.

Selecting the Default Startup Set Using Basler pylon

The User Set Default Selector is used to select the default startup set:

- Set the User Set Default Selector to User Set 1, User Set 2, User Set 3, or Default.

You can set the User Set Default Selector from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector:

```
// Designate user set 1 as the startup set
Camera.UserSetDefaultSelector.SetValue( UserSetDefaultSelector_UserSet1 );

// Designate the default set as the startup set
Camera.UserSetDefaultSelector.SetValue( UserSetDefaultSelector_Default );
```

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 3.1 on page 15.

Selecting the Default Startup Set Using Direct Register Access

When using direct register access, the User Set Default Selector register is used to select the startup set:

- Set the value of the User Set Default Selector register for User Set 1, User Set 2, User Set 3, or Default as desired.

For more information about direct register access, see Section 3.2 on page 16.

9 Troubleshooting and Support

This chapter outlines the resources available to you if you need help working with your camera.

9.1 Tech Support Resources

If you need advice about your camera or if you need assistance troubleshooting a problem with your camera, you can contact the Basler technical support team for your area. Basler technical support contact information is located in the front pages of this manual.

You will also find helpful information such as frequently asked questions, downloads, and application notes in the Downloads and the Support sections of our website:
www.baslerweb.com

If you do decide to contact Basler technical support, please take a look at the form that appears on the last two pages of this section before you call. Filling out this form will help make sure that you have all of the information the Basler technical support team needs to help you with your problem.

9.2 Obtaining an RMA Number

Whenever you want to return material to Basler, you must request a Return Material Authorization (RMA) number before sending it back. The RMA number **must** be stated in your delivery documents when you ship your material to us! Please be aware that if you return material without an RMA number, we reserve the right to reject the material.

You can find detailed information about how to obtain an RMA number in the Support section of our website: www.baslerweb.com

9.3 Before Contacting Basler Technical Support

To help you as quickly and efficiently as possible when you have a problem with a Basler camera, it is important that you collect several pieces of information before you contact Basler technical support.

Copy the form that appears on the next two pages, fill it out, and fax the pages to your local dealer or to your nearest Basler support center. Or, you can send an e-mail listing the requested pieces of information and with the requested files attached. Basler technical support contact information is shown in the title section of this manual.

1 The camera's product ID:

2 The camera's serial number:

3 Network adapter that you use with the camera:

4 Describe the problem in as much detail as possible:
(If you need more space, use an extra sheet of paper.)

5 If known, what's the cause of the problem?

6 When did the problem occur? □ After start. □ While running.
□ After a certain action (e.g., a change of parameters):

7 How often did/does the problem occur?

□ Once. □ Every time.

□ Regularly when:

____

□ Occasionally when:

____

____

____

____

8 How severe is the problem?

□ Camera can still be used.

□ Camera can be used after I take this action:

____

____

□ Camera can no longer be used.

9 Did your application ever run without problems?

□ Yes □ No

10 Parameter set

It is very important for Basler technical support to get a copy of the exact camera parameters that you were using when the problem occurred.

To make note of the parameters, use Basler’s pylon Viewer tool.

If you cannot access the camera, please try to state the following parameter settings:

□ Frame Size: ______________________________

□ Pixel Format: ______________________________

□ Packet Size: ______________________________

□ Exposure Time: ______________________________

□ Line rate: ______________________________

11 Live image/test image

If you are having an image problem, try to generate and save live images that show the problem. Also generate and save test images. Please save the images in BMP format, zip them, and send them to Basler technical support.

Revision History

Doc. ID Number	Date	Changes
AW00118501000	13 June 2012	Preliminary release of this document. Applies to prototype cameras only.

Index